

Metropolia Ammattikorkeakoulu
Mediatekniikan koulutusohjelma

Ilmari Oranen

Www-pohjaisen videojulkaisujärjestelmän arkkitehtuuri

Insinööritö 2.6.2010

Ohjaaja: kehityspäällikkö Jonas Kronlund

Ohjaava opettaja: yliopettaja Kari Aaltonen

Tekijä Otsikko Sivumäärä Aika	Ilmari Oranen Www-pohjaisen videojulkaisujärjestelmän arkkitehtuuri 67 sivua 2.6.2010
Koulutusohjelma	mediatekniikka
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	kehityspäällikkö Jonas Kronlund yliopettaja Kari Aaltonen
<p>Insinööriyöprojektin tarkoituksena oli kehittää yhteisöllisen ja monikanavaisen videopalvelun tekninen arkkitehtuuri ja sen toteutus. Tarkemmat tekniset tavoitteet muuttuivat kuitenkin jatkuvasti projektin aikana. Työssä paneuduttiin erityisesti www-palvelinarkkitehtuuriin ja Drupal-nimisen sisällönhallintajärjestelmän hyödyntämiseen. Insinööriyöraportissa tarkastellaan myös yleisemmin Internet-pohjaista videotekniikkaa ja vertaillaan muiden samantyyppisen projektien toteutuksia ja alan mahdollisia suuntauksia.</p> <p>Toteutetun palvelun arkkitehtuurin taustalla olevat ohjelmistoteknologiat ovat enimmäkseen ilmaisia vapaan lähdekoodin sovelluksia. Niihin päädyttiin osittain tiukan budjetin sanelemana, mutta ensisijaisesti muista syistä. Ohjelmistomaailmassa löytyy lähes kaikelle ilmaisia vaihtoehtoja, jotka ovat usein huomattavasti kaupallisia tuotteita laajemmassa käytössä ja välillä myös käyttötarkoitukseensa ylivertaisia.</p> <p>Flash-soittimen ja H.264-videokoodekin monopoliasemat saattavat olla uhattuina lähitulevaisuudessa. Ne voivat joutua avoimempien ja maksuttomien teknologioiden, kuten HTML5-standardin ja Googlen juuri hankkiman VP8-koodekin, taakse.</p> <p>Varsinaisen pääprojektin, eli yhteisöllisen ja monikanavaisen videopalvelun, teknisen toteutuksen tuloksista voidaan päätellä, että käytössä olleilla teknisillä työkaluilla voidaan tarvittaessa tuottaa rajoitettujen pilottien lisäksi myös suuremman kokoluokan kaupallisia ratkaisuja. Tähän vaaditaan kuitenkin aina kunnon resursseja, pääasiassa paljon henkilötyötunteja ja hyvässä tasapainossa olevat suunnittelu- ja toteuttamisryhmät. Tusina visioijaa yhtä toteuttajaa kohti ei ole sopiva suhde. Ohjelmointiprojekteissa pitäisi myös osata keskittyä ominaisuuksia suunniteltaessa asian ytimeen ja koettaa hillitä liiallista rönsyilemistä.</p>	
Hakusanat	julkaisujärjestelmä, sosiaalinen media, www, video, Drupal, VP8

Author	Ilmari Oranen
Title	The infrastructure of an online video publishing platform
Number of Pages	67 pages
Date	2.6.2010
Degree Programme	Media Technology
Degree	Bachelor of Engineering
Instructor Supervisor	Jonas Kronlund, Development Manager Kari Aaltonen, Principal Lecturer
<p>This thesis describes the infrastructure of an online video publishing platform, and the processes through which it was created. It is concentrated on the details of web-server setup and the content management system Drupal, which is used to integrate the different parts together.</p> <p>The results suggest that an infrastructure like this can mostly be built with free, open source tools. The Drupal-framework can be successfully utilized in creating fast developing pilot sites, as well as massive full blown commercial portals.</p> <p>Considering the results of the writer's experiences, it was learned that it is best to concentrate on a few key aspects instead of trying to implement all and any feature-requests. This is especially true when working with very limited resources.</p> <p>Also, it seems the industry standards, Flash video and the H.264-video codec may soon be surpassed by the HTML5-standard's inbuilt video player and Google's recent acquisition: the VP8-video codec.</p>	
Keywords	Cms, web, video, Drupal

Sanasto

CMS (*Content Management System, sisällönhallintajärjestelmä, julkaisujärjestelmä*)

Järjestelmä, jolla hallitaan erimuotoista tietoa. Esimerkiksi www-pohjainen asiakasrekisteri, kuvapankki, artikkelitietokanta, tai näiden yhdistelmä.

Contributed module

Laajennusmoduuli (esimerkiksi Drupalille), jolla palvelun ominaisuuksia voidaan laajentaa ilman omaa ohjelmointia.

Core module

Järjestelmän perusasennuksen mukana tulevat moduulit.

Cron

Unix-järjestelmissä käytettävä apuohjelma, joka suorittaa käskyjä ja muita ohjelmia määritetyllä aikavälillä.

DVB-H (Digital Video Broadcasting - Handhelds)

Digitaalisten televisiolähetysten standardi mobiililaitteille.

HTTP (Hypertext Transfer Protocol)

Www-selaimien käyttämä yleinen protokolla, joka perustuu pyyntöihin (POST/GET request) ja vastauksiin (response).

Klusteri

Klusteri tarkoittaa usean palvelimen näkymistä verkkoon yhtenä palvelimena. Usein palvelimet ovat fyysisesti toisiaan lähellä ja yhdistetty lähiverkolla.

Memcache

Memcache on useaa yhdistettyä palvelinta tukeva välimuistitekniikka, jolla pyritään minimoimaan turhat kyselyt tietolähteisiin. Järjestelmä on käytössä lukuisilla Facebookin ja YouTube'n kaltaisilla erittäin suuren kokoluokan sivustoilla.

NFS (Network File System)

NFS on Linux-järjestelmän tapa jakaa levyosioita verkon välityksellä.

Patch

Aktivoitava koodimuutos sovelluksen laajentamiseksi tai ohjelmavirheen korjaamiseksi.

RTMP (Real Time Messaging Protocol)

Protokolla Flash-multimediasisältöjen lähettämiseen palvelimelta käyttäjän toistimeen. Toimii TCP-, HTTP- tai HTTPS-protokollan kautta.

Symlink (Symbolic link)

Symbolinen linkki (eli symlinkki) on Unix-pohjaisen järjestelmän (kuten Linux) tiedostopuussa sijaitseva viittaus toisaalle tiedostorakenteessa.

Virtuaalipalvelin

Käyttöjärjestelmä, joka toimii itsenäisesti toisen käyttöjärjestelmän sisällä. Tällä tekniikalla voidaan luoda samalle fyysiselle palvelimelle esimerkiksi useita toisistaan erillisiä kehitysympäristöjä ja www-palveluja.

VOD (Video-On-Demand, tilausvideo)

Video, joka ladataan katselua varten (yleensä Internetistä). Lähetys on epälineaarinen, koska vastaanottajat voivat katsoa sen haluamanaan hetkenä.

Sisältö

Tiivistelmä

Abstract

Sanasto

1 Johdanto.....	8
2 Videojulkaisujärjestelmät	9
2.1 Www-pohjaisten videoteknologioiden kehitys	9
2.3 Videoiden käsittelyprosessi	13
2.4 Videoiden esittäminen	15
3 Teknisen arkkitehtuurin vaatimukset.....	17
3.1 Ohjelmointiympäristö.....	17
3.2 Arkkitehtuuriin kuuluvat järjestelmät	20
3.3 Järjestelmien yhdistäminen.....	20
3.4 Monikanavaisuus	23
3.5 Multimediatiedostojen käsittely	26
3.6 Pullonkaulat ja resurssien hallinta.....	27
3.7 Skaalautumismahdollisuudet	30
4 Ulkoisten järjestelmien integrointi Drupal-alustaan	31
4.1 Alustan soveltuvuus.....	31
4.2 Avoimen lähdekoodin työkalut	33
4.3 Kaupalliset työkalut	37
4.4 Modulaarinen rakenne	39
4.5 Sosiaalisten mediasivustojen rajapintojen hyödyntäminen.....	40
4.6 Käyttäjätasot ja käyttöoikeudet	42
5 StadiTV-palvelun testaus.....	43
5.1 Pilotti: videoiden lähetys mobiililaitteilla	43
5.2 Käyttöliittymän kehitys ja testaus	45
5.3 Ohjelmoinnin laadunvalvonta	46

5.4 Tietoturvariskien hallitseminen.....	47
6 Vertailu muihin saman alustan videojärjestelmiin.....	48
6.1 Omat projektit	48
6.2 Suomalaisia videopalveluita	55
7 Yhteenveto	57
Lähteet	59
Liite 1: Multimediasisältöjen käsittelyvaiheet.....	63
Liite 2: Luettelo käytetyistä Drupal-moduuleista	64
Liite 3: Sivuston käyttöliittymän eri versiot.....	65

1 Johdanto

Insinööritö sai alkunsa marraskuussa 2008 Jonas Kronlundin (kehityspäällikkö, *Elisa Oyj*) kysyessä kiinnostustani osallistua yhteisöllisen videojulkaisupalvelun, eli StadiTV:n, kehittämiseen. Alustaksi ehdotettiin *Drupal*-nimistä sisällönhallinta-järjestelmää. Olin päätenyt työskentelemään Elisa Oyj:n *Tutkimusyhteistyö*-ryhmään toukokuussa 2007, ja ensimmäisestä työviikosta asti olin toteuttanut erilaisia projekteja Drupal-alustalla. Alun perin päädyinkin käyttämään Drupalia muun muassa Jonas Kronlundin suositeltua sitä ensimmäiseen projektiin.

Sitten olen käytännössä ansainnut elantoni Drupal-projekteilla, ja nähtyäni alustan nopeasti kiihtyvän kasvun viime vuosina, olen päättänyt pysyä samalla uralla. Halusin myös tehdä lopputyöni Drupal-projektista, ja vaihtoehtoisia tekijöitä tarpeellisella erikoisosaamisella ei ollut tarjolla, joten aiheen valinta oli selvä.

Insinööritöön tavoitteeksi muodostui palvelun kehitysvaiheen tekninen suunnittelu ja toteutus. Projekti toteutettiin yhdessä Forum Virium -yhteisön kanssa. Mukana oli muun muassa Arcada-ammattikorkeakoulu visioimassa, mitä palvelu voisi sisältää ja millä ansaintamallilla se voisi tulla kannattavaksi. Oma näkökulmani projektiin on kuitenkin puhtaasti tekninen, ja insinööritössä keskitytään vain teknisiin osa-alueisiin.

Työn painopisteisiin kuului alun perin myös *monikanavaisuus*: video on demand, lineaarinen lähetys sekä mobiili-, www- ja kaapeliversio. Ominaisuuslistan paisuessa jatkuvasti päätin lopulta kuitenkin keskittyä insinööritössä www-puoleen, koska en työnantajan vaihtumisen takia ehtinyt toteuttaa kuin osan monikanavatoiminnoista. Pääprojektin viimeistelemissä myöskin insinööritöraportissa tarkastellaan vertailukohteena myös muutamaa muuta saman julkaisujärjestelmän avulla toteutettua samantyyppistä www-projektia, joissa olen ollut osallisena.

2 Videojulkaisujärjestelmät

2.1 Www-pohjaisten videoteknologioiden kehitys

Internetissä toimivat videopalvelut eivät ole uusi keksintö. 1990-luvun hitaista modeeminopeuksista huolimatta tarjontaa ja kysyntää riitti silloinkin kaikentyyppisille mediasisällöille, aikuisviihde mukaan lukien. Tiedostokoot ja kuvanlaatu olivat tekniikan rajoitteiden takia nykyisiin verrattuina varsin alhaisia. Hitaiden tiedonsiirtonopeuksien takia videot olivat suoratoiston sijaan useimmiten tarkoitettu ladattavaksi ensin kiintolevyille. Varsinaista *streamaus*-tekniikkaa (suoratoisto) alettiin hyödyntää enemmän tiedonsiirto- ja prosessointitehojen kehittyessä. [1.]

Eri tarkoituksiin optimoituja video- ja audioformaatteja sekä -koodekkeja on ehtinyt kehittyä ja vanhentua runsaasti. Www-pohjaiseen suoratoistoon tarkoitettujen formaattien kehittyessä nopeasti on myös suosituin formaatti muuttunut jatkuvasti. Esimerkiksi Microsoftin *wmv*, Applen QuickTime (*mov*) ja RealMedian *rm* menettivät vuoroin valta-asemansa. [2; 3.]

Videotiedoston tiedostopääte (esim. *avi*, *ogm*, *mkv*, *mp4*) ei välttämättä kerro, millä koodekeilla sen ääni- ja videoraidat on pakattu. Monista suosituista koodekeista on sekä kaupallisia että maksuttomia lisenssivapaita variantteja. Kaupallisiin koodekkeihin liittyy erittäin monimutkaisia patenttioikeuksia, joista käydään jatkuvasti lakitaisteluja [4.]. Vanhentuneet tekniikat eivät kuitenkaan liity oleellisesti insinööriyön ydinalueeseen, joten niitä ei käsitellä tämän enempää.

Nykyinen videostandardi: H.264

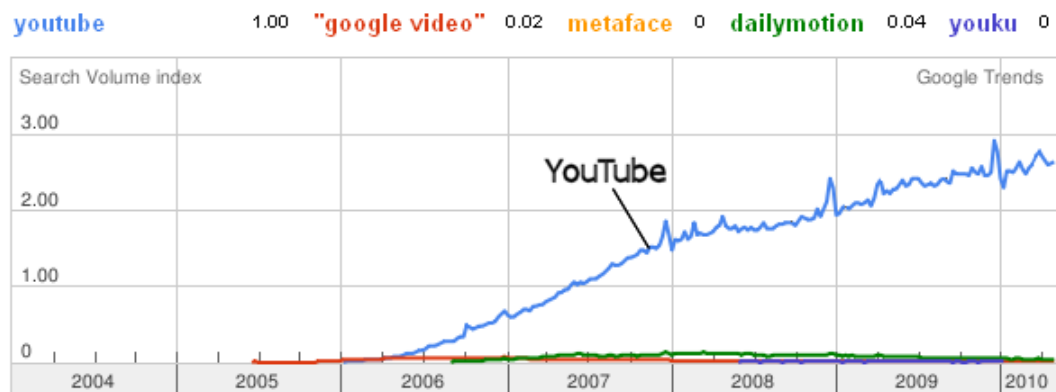
Nykyisin selkeästi suosituin ja laajimmin tuettu videostandardi on *H.264 (MPEG-4/AVC)*. Se määrittelee teknisesti useita eritasoisia profiileja eri käyttötarkoituksiin ja

on saavuttanut toistaiseksi parhaiten tuetun standardin aseman ainakin Internetin suoratoistovideoille, mobiililaitteille, digi-tv-lähetyksille ja Bluray-videoille. Standardille on olemassa useita suljettuja kaupallisia koodekkeja ja yksi maksuton avoimen lähdekoodin toteutus: *x264*. [5; 6.]

Videoteknologioiden menestymisen kannalta on ensiarvoisen tärkeää, että tekniikka on käytettävissä kaikilla yleisillä käyttöjärjestelmillä. H.264-videoiden pakkaus (*encoding*) ja toisto (*decoding*) onnistuu Windowsilla, Linuxilla, Mac OS:lla ja lukuisilla harvinaisemmilla päätelaitteilla. Korkealaatuisten videoiden käsitteleminen rasittaa prosessoria huomattavasti, mutta standardin yleistyessä useat grafiikkakorttien valmistajat ovat lisänneet H.264-tuen suoraan komponentteihinsa, mikä vähentää prosessorin taakkaa. [5.]

Johtavat palvelut

Palveluiden suosiota voidaan mitata monin tavoin, kuten hakuvolyymien, kävijöiden, lisättyjen videoiden, katsottujen videoiden, tiedonsiirtokaistan tai liikevaihdon mukaan. Vuonna 2005 aloittanut *YouTube* on vakiinnuttanut videomarkkinoilla ylivoimaisen aseman. Kuva 1 osoittaa YouTube'n hakuvolyymien monikymmenkertaiseksi lähimpiin haastajiin verrattuna.



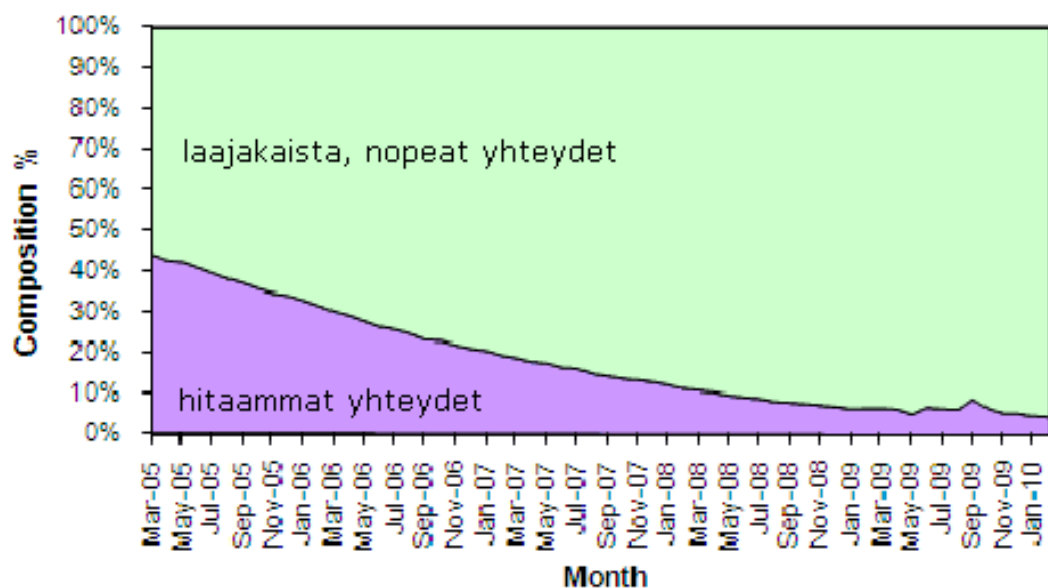
Kuva 1. Muutamien tunnettujen videopalvelujen hakumäärien vertailu vuosina 2004–2010. YouTube erottuu lähimmistä kilpailijoistaan, koska se on jatkuvasti kasvanut ja sen hakuvolyymi on monikymmenkertainen. [8; 9.]

Wikipedian videosivustojen vertailu osoittaa samoja voimasuhteita myös muilla vertailumittareilla. Suosiosta huolimatta YouTube tekee toistaiseksi vielä tappiota. Sen on kuitenkin arvioitu muuttuvan voitolliseksi lähitulevaisuudessa. [7; 8; 9; 10.]

Koska YouTube on vakiinnuttanut paikkansa alan huipulla, sitä käytetään tässä raportissa paikoin teknisenä vertailukohtana.

Videopalveluiden läpimurto ja laajakaista

Internetissä toimivien videopalveluiden suosion kasvu liittyy vahvasti laajakaistojen yleistymiseen. Kuten *kuva 2* osoittaa, vuotta 2005 voidaan pitää läpimurtovuotena, sillä silloin laajakaistaa käytti ainakin Yhdysvalloissa jo yli puolet aktiivisista Internetin käyttäjistä.



Kuva 2: Laajakaistan käyttöaste aktiivisten web-käyttäjien talouksissa Yhdysvalloissa vuosina 2005–2010 välillä [11].

Myös YouTube aloitti toimintansa vuonna 2005. Lisäksi samana vuonna *Flash 8* -videotoistin sai tuen *On2*-yrityksen kehittämälle entistä tehokkaammalle videokoodikille nimeltä *VP6*. Vuonna 2007 YouTube siirtyi käyttämään *H.264*-koodekkia pian sen jälkeen kun *Flash* alkoi tukea sitä. [12; 10.]

Tiedonsiirtokaistan halventumisesta huolimatta ovat suurten videopalveluiden tiedonsiirtokulut nousseet jatkuvasti palvelujen kasvun ja teräväpiirtovideoiden suosion myötä. YouTube on osittain ulkoistanut suosituimpien videoidensa lähettämisen tehokkaisiin mutta maksullisiin *CDN*-palveluihin (*Content Delivery Network*). [8.]

Mahdollinen tulevaisuus: avoin videoformaatti

H.264 ei ole vapaan lähdekoodin alainen, vaan lisensoitu tuote. Käytöstä ei toistaiseksi vaadita lisenssimaksuja ainakaan vuoteen 2016 asti, mutta niiden on pelätty tulevan käyttöön tulevaisuudessa. Useat tahot ovat huolissaan mahdollisista tulevista lisenssikiistoista. *H.264* alkaa olla niin vakiintunut, että siitä olisi vaikea päästä eroon, vaikka lisenssimaksut nousisivat kiusallisen korkeiksi. [4; 13.]

YouTuben omistava *Google* on ryhtynyt toimenpiteisiin, joilla on todennäköisesti tarkoitus luoda oma videostandardi, joka syrjäyttäisi *H.264*:n. Google osti *On2*-yrityksen helmikuussa 2010 ja aikoo eräiden lähteiden mukaan julkaista yrityksen *VP8*-koodekin avoimen lähdekoodin lisenssillä toukokuussa 2010. Tämä saattaa uhata kaupallisen *H.264*-formaatin asemaa ja mullistaa videojulkaisualan jälleen kerran, ainakin Internetin osalta. Mikäli näin tapahtuu, *www*-teknologioiden yleisen kehityksen kannalta olisi erittäin tärkeää, että mahdollinen uusi standardi olisi täysin avoin ja lisenssitön. [14; 15; 16; 17.]

Flashin korvaaja

Google pyrkii oletettavasti suljetun videokoodin lisäksi eroon myös suljetusta videosoittimesta eli *Flash Player*istä. Flash-teknologian mahdollinen korvaaja voisi tulla HTML5-standardin mukana. HTML5-standardi on vasta hyvin varhaisella kehitystasolla, mutta se laajentaa perusselaimen ominaisuuksia muun muassa natiivilla videosoittimella. HTML5-pohjainen videotoistinstandardi on jo otettu testikäyttöön ainakin *Firefox*- ja *Chrome*-selaimissa, ja sekä YouTubeille että sen kilpailijalle *Vimeolle* on olemassa HTML5-testiversio videoiden selailuun. Videot käyttävät toistaiseksi H.264-koodia, mutta pidetään todennäköisenä, että YouTube siirtyy tulevaisuudessa omistamansa VP8-koodin käyttöön. [16; 17; 18; 19.]

Mahdollinen videostandardin vaihtuminen ei kuitenkaan tapahtuisi kovin nopeasti, joten H.264/Flash-yhdistelmä oli ainoa järkevä valinta myös StadiTV-projektin videoformaattiksi.

2.3 Videoiden käsittelyprosessi

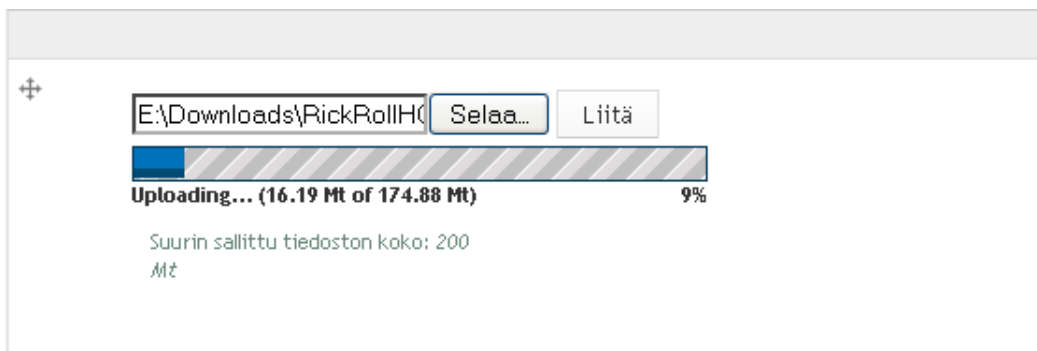
Videotiedostojen siirto

Videotiedostot siirretään videojulkaisujärjestelmissä www-palvelimelle normaalisti suoraan selaimen käyttämän HTTP(S)-protokollan avulla. Käyttäjä liittää tiedoston lomakkeeseen, joka lähetetään POST-pyyntönä vastaanottavalle palvelimelle.

Yleensä liitetiedoston siirto ei tuota www-palvelun sivuston rakentamisen kannalta merkittäviä lisähaasteita. Videopalveluissa tiedostot ovat kuitenkin hyvin suuria, ja käyttäjät saattavat kyllästyä hitaaseen tiedonsiirtoon, mikäli eivät näe visuaalista palautetta tiedonsiirron etenemisestä. Aikaisemmin tämän tiedon esittämiseen PHP:lla tuotetulla sivustolla vaadittiin jokin vaihtoehtoinen tekniikka HTML-lomakkeen sijaan,

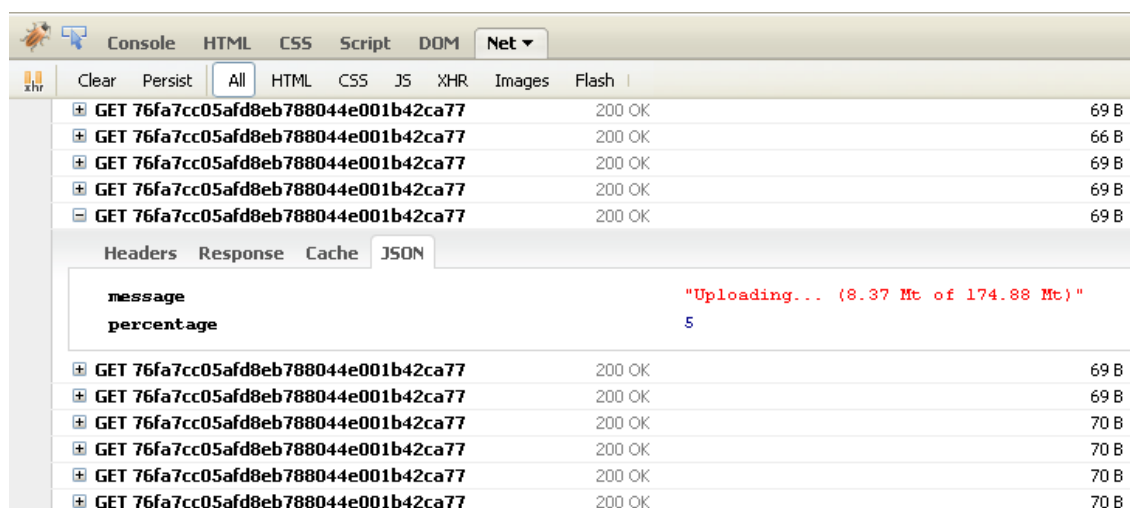
mutta nykyään tiedonsiirron saa visualisoitua kohtuullisella vaivalla Apachen *uploadprogress*-laajennuksella. Laajennuksen voi aktivoida *PECL*-kirjaston avulla.

Drupalin *filefield*-moduuli tukee *uploadprogress*-toiminnallisuuksia suoraan ilman mitään lisäasetuksia. Kuva 3 osoittaa, miltä tiedonsiirto näyttää käyttäjälle.



Kuva 3: Latauspalkki näyttää tiedonsiirron etenemisen käyttäjälle graafisesti ja numeerisesti.

Tämä reaaliaikainen palaute hyödyntää AJAX-tekniikkaa (*Asynchronous JavaScript and XML*). Selain lähettää noin kahden sekunnin välein AJAX-pyyntöjä palvelimelle. Kuhunkin pyyntöön saadaan vastauksena pieni JSON-muotoinen tekstuaalinen (message) ja prosentuaalinen (percentage) tieto tiedonsiirrosta. Kuva 4 esittää AJAX-pyyntöt Firefox-selaimen *Firebug*-laajennuksella nähtynä.



Kuva 4: Firebug havainnollistaa *uploadprogress*-laajennuksen tuottamat AJAX-pyyntöt.

Muuntamisprosessi

StadiTV:ssä videotiedoston muuntaminen toistossa käytettävään H.264-formaattiin ei tapahdu suoraan POST-siirron yhteydessä. Sen sijaan palvelimella suoritetaan kolmen minuutin välein cron-ajo, joka käy läpi uudet videosivut ja näiden liitteet. Drupal suorittaa sitten *flashvideo*-moduulissa asetuksissa määritetyn FFmpeg-sovelluksen komennon, joka luo originaalitiedostosta uuden version. Luotu tiedosto käyttää H.264-videokoodekkia ja AAC-äänikoodekkia.

StadiTV:ssä käytettävät tarkemmat tiedot FFmpeg-konversiosta selitetään luvussa 4.2. Videoiden käsittely FFmpeg:llä suoraan www-palvelimella on käytössä lukuisissa kaupallisissakin palveluissa, mutta alan suurimmissa palveluissa videokonversion pitäisi aina tapahtua erillisellä palvelimella, ettei raskas videokonversio hidasta www-palvelua.

2.4 Videoiden esittäminen

Videosoitin

Videoiden esittämiseen käytetään Internetin videosivustoilla pääasiassa Flash-tekniikkaa, koska se toimii lähes kaikkien käyttäjien selaimissa. Uusin versio (*Flash 10*) ei tosin toimi toistaiseksi Linux-laitteilla. Myöskään Applen iPhone- ja iPad-mobiililaitteet eivät tue sitä. Useimmille mobiililaitteille on kuitenkin olemassa karsitut Flash Lite -versiot, joita voi käyttää myös videoiden toistoon. [12.]

StadiTV-projektiin hankittiin soittimeksi suosittu *JW FLV Player*. Luvussa 4.3 kerrotaan tarkemmin soittimen käyttöönotosta. Tulevaisuudessa Flash-soittimen saattaa kuitenkin siis syrjäyttää selaimen natiivi HTML5-määritystä tukeva videoitoin.

Progressiivisesti ladattavat videot

Flash-videoiden lähetystapana voidaan yksinkertaisimmillaan käyttää progressiivista lataustamenetelmää. Sen käyttö ei vaadi mitään ylimääräisiä palvelinsovelluksia tai asetuksia. Tiedonsiirto toimii kuten muukin www-palvelimen välittämä tapahtuma, jolloin Flashin videontoistoon käyttämä protokolla toimii HTTP-protokollan pakettien alaisuudessa.

Progressiivisessa latauksessa tiedosto haetaan aina kokonaisuudessaan selaimen muistiin, vaikka käyttäjä poistuisi sivulta. Videota voi kelata sen verran kuin se on latautunut, mutta loppuosaan ei voi siirtyä lataamatta ensin koko tiedostoa. Videon kaistanleveyttä ei voi myöskään muuttaa kesken toiston.

Laajemmassa käytössä tämä streamaustapa tuhlaa jonkin verran palvelimen ja käyttäjän resursseja, minkä takia merkittävät videopalvelut käyttävät progressiivisen latauksen sijasta tehokkaampaa suoratoisto-tekniikkaa, joka vaatii enemmän palvelinarkkitehtuurilta. StadiTV:n osalta viimeisimmässä testiversiossa on kuitenkin vielä käytössä progressiivinen lataus testauksen helpottamiseksi.

Suoratoistovideon lähettäminen

Todellisessa Flash-palvelimen tuottamassa streamissa ladataan aina vain tarvittava osa videosta. Jos käyttäjä haluaa nähdä vain videon lopun, haetaan vain loppu. Tämä kuitenkin vaatii *RTMP*-protokollan (Real Time Messaging Protocol) streamauskäyttöön kykenevän *Adobe Flash Media Server* -sovelluksen tai jonkin sen avoimista varianteista, kuten *Red5* tai *Wowza Media Server*. Flash-soitin pääsee tällöin kommunikoimaan suoraan videopalvelimen kanssa käyttämällä täysin omaa TCP-

pohjaista protokollaa. Protokollasta on olemassa myös RTMPS-versio, jolloin tiedonsiirto tapahtuu salatun HTTPS-yhteyden läpi.

[20; 21.]

Flash-palvelin jakaa haettavan audio- ja videoraidan erilleen ja päättää lähetettävien pakettien koon muun muassa käyttäjän tiedonsiirtonopeuden mukaan. Videota aletaan lähettää pala kerrallaan soittimelle, joka kokoaa palat näytettäväksi oikeassa järjestyksessä. Tämä mahdollistaa myös reaaliajassa kuvatun lähetyksen seuraamisen.

3 Teknisen arkkitehtuurin vaatimukset

3.1 Ohjelmointiympäristö

Ohjelmointikielen ja -alustan valitseminen

Projektin alusta asti oli selvää, että se toteutettaisiin *PHP*-ohjelmointikielellä. Syynä on ensisijaisesti työn tekijän noin 6 vuoden kokemus kielestä ja henkilökohtainen mieltymys, jonka mukaan PHP (*Hypertext Preprocessor*) soveltuu tämänkaltaisiin www-projekteihin paremmin kuin kilpailevat teknologiat. Www-ohjelmointikielten ylivertaisuudesta löytyy runsaasti eriäviä mielipiteitä Internetistä, mutta tasokkaimmissa keskusteluissa on useimmiten johtopäätöksenä, että parhaan työkalun valinta riippuu aina työn vaatimuksista, ja tekijän teknisestä suuntautumisesta. [22; 23; 24.]

Projektiin suunniteltujen ominaisuuksien takia ei olisi ollut järkevää yrittää ohjelmoida kaikkea alusta asti, vaan hyödyntää olemassa olevia ohjelmistoja, joita integroimalla ja muokkaamalla säästetään aikaa ja resursseja. Tämän takia kannattaa ottaa pohjaksi jokin tunnettu ja hyvin laajennettavuutta tukeva järjestelmä. Tätä ”älä keksi pyörää uudelleen” -ideologiaa käytetään yleisesti lähes kaikissa ohjelmistoprojekteissa.

Järjestelmän ytimessä toimivaksi alustaksi valittiin sisällönhallintajärjestelmä nimeltä *Drupal*. Alusta on luotu PHP:lla, minkä takia siihen on kätevintä myös integroitua samalla kielellä toteutetuilla moduuleilla. Tämä ei toki estä muiden ohjelmointikielien hyödyntämistä, joskin niiden yhdistäminen samaan järjestelmään on paljon työläämpää.

Muita vaihtoehtoja ohjelmointikieleksi olisivat olleet esimerkiksi Java, .NET (ASP, C#), joista tekijällä on kokemusta lähinnä opiskeluprojekteista. Kokemuksen vähäisyydestäkin huolimatta niille ei olisi löytänyt projektin tarpeisiin sopivaa Drupalin kaltaista, joustavaa ja tehokasta vapaan lähdekoodin alustaa. Pikemminkin työn tekijä valittiin työn vaatimusten mukaan eikä toisinpäin.

Käyttöjärjestelmän valitseminen

Palvelimen käyttöjärjestelmäksi on nykyään mahdollista valita lähinnä Windows tai Linux. Teoriassa myös UNIX-pohjainen Mac OS voi toimia palvelinkäytössä, mutta harvemmin sitä edes harkitaan. Suosituimpina voidaan pitää Linuxin eri variantteja. Kaikille mainituille järjestelmille on saatavilla kaikki projektin tarpeisiin vaadittavat lisäohjelmistot tietokannasta videonkäsittelyyn.

Linuxin vahvuuksiin palvelinkäytössä kuuluvat turvallisuus, avoimuus, luotettavuus ja edullisuus. Myös Microsoftin Windowsilla on merkittävä asema palvelinmarkkinoilla, mutta Linux näyttäisi pysyvän vielä selkeässä johdossa, toisin kuin desktop-tietokoneissa. Windows-palvelimien eduksi voidaan mainita kehittynyt tuotetuki ja käyttäjäystävällisempi graafinen käyttöliittymä. Palvelimia tosin hallitaan lähinnä komentoriviltä tai www-käyttöliittymistä.

Linux-ohjelmistot ovat itse käyttöjärjestelmän tavoin lähes kaikki avoimen lähdekoodin tuotoksia ja valtaosin myös ilmaisia, mitä voidaan pitää syynä järjestelmän laajempaan levinneisyyteen erityisesti palvelinkäytössä. Kriittisen käyttäjämässän saavuttaminen on

johtanut ongelmanratkaisujen helpottumiseen. Kun järjestelmällä on tarpeeksi käyttäjiä, jotka ilmoittavat teknisistä ongelmistaan Internetin keskustelupalstoille, löytyy niihin hakukoneiden avulla yleensä myös asiantuntevia vastauksia, kuten toki kaupallisillekin tuotteille. Vastausten tulkitseminen tosin vaatii yleensä jonkin verran teknistä tuntemusta ja kielitaitoa.

Avoin lähdekoodi ei tarkoita aina ilmaisuutta, ja projektin käyttöjärjestelmäksi valikoituikin maksullinen RedHat Linux. Kyseessä on nimen omaan ammattimaiseen palvelinkäyttöön suunniteltu järjestelmä, johon kuuluu myös täysi tuotetuki. Samalla käyttöjärjestelmällä oli *ElisaLabs*-ryhmässä ennestään toteutettu virtuaalipalvelimien avulla erilaisia www- ja mobiilipalveluita ja niiden pilotointiprojekteja. StadiTV-projektin laskettiin kuitenkin vaativan itsenäisen projektille omistetun palvelimen, joten sitä ei virtualisoitu. [25.]

Mikäli palvelu saavuttaisi suuren suosion, pitäisi järjestelmä pystyä jakamaan usealla palvelimella. Arkkitehtuurilta ei tässä vaiheessa vaadittu merkittävää skaalautuvuutta, mutta skaalautuvuuden mahdollisia toteutustapoja käsitellään luvussa 3.7.

Palvelimen käyttöjärjestelmän lisäksi on valittava myös itse www-palvelinohjelmisto, joka hoitaa dynaamisten www-sivujen generoimisen ja tiedostojen siirrot ja vastaa lähinnä *HTTP*-protokollaan kuuluvista toimenpiteistä. Oikeastaan valinta tehtiin jo käyttöjärjestelmän mukana, sillä Apachen www-palvelinsovellus on selkeästi suosituin vaihtoehto Linuxille. Apache toimii kaikilla alustoilla, toisin kuin *IIS* (Internet Information Services), joka olisi ollut Windows-ympäristössä todennäköisin valinta. Drupal tukee molempia, mutta sen asennusohje suosittelee selkeästi Apachen käyttöä. *IIS*:n käyttöönotto Drupalin kanssa olisi vaatinut käyttöjärjestelmän vaihtamisen lisäksi huomattavasti erikoissäätöä, ilman mitään etuja. [26.]

Ammattimaisessa www-palvelinkäytössä on myös muita vaihtoehtoja, kuten *lighttpd* ja *nGinx*, jotka ovat ominaisuuksiltaan karsituimpia, mutta suunniteltu skaalautumaan

erityisesti suurille käyttäjämassoille. Apache on kuitenkin tuttu ja laajimmin tuettu, ollen oikein konfiguroituna myös huomattavan tehokas. [27; 28.]

Lisäksi oli valittava myös tietokantajärjestelmä. Järjestelmäksi ei harkittu mitään muuta kuin MySQL-tietokantaa, joka on työn tekijälle ehdottomasti tutuin ja Drupalin parhaiten tukema tietokantatyyppe. Drupal tukee myös PostgreSQL-tietokantaa, mutta toisin kuin MySQL, se ei ole yhteensopiva kaikkien moduulien kanssa.

3.2 Arkkitehtuuriin kuuluvat järjestelmät

Arkkitehtuurin haluttiin saada monikanavaiset videotoinnot: videostreamausta mobiililaitteille Internetissä tai *DVB-H*-signaalin (mobiili-tv) avulla. Suunnitelmana oli käyttää tähän valmiita työkaluja: Oulun seudun ammattikorkeakoulun kehittämää *Neo Arena TvExperience* -alustaa Internetin suoravideoistoon ja kaupallista *Sofia Digital Backstagea* mobiili-tv-versiota varten. TvExperienceen toimintoihin kuuluu videoiden siirtäminen palveluun, sisällön luokittelu ”kanaviin” ja niiden esittäminen selaimessa.

Neo Arena toimii yhteistyössä Elisan kanssa ja luovutti TvExperience-järjestelmänsä lähdekoodin ja dokumentoinnin Elisan käyttöön. Järjestelmään tutustumisen jälkeen todettiin, että toiminnot voidaan siirtää Drupal-moduuliin.

3.3 Järjestelmien yhdistäminen

Neo Arena TvExperience

TvExperience on pieni ja selkeästi ohjelmoitu PHP/MySQL-sovellus, joka on tehty *pureMVC*-alustalle. Sen pohjalta pyrittiin luomaan Drupal-moduuli, joka hyödyntäisi Drupalia yleisiin toimintoihin, kuten kirjautumiseen, käyttöoikeuksiin, ylläpitoasetuksiin, ulkoasuun ja sisällön syöttöön sekä luokitteluun.

Oikeastaan moduulin ainoaksi toiminnoksi jäisi tällöin videoiden konversio ja esittäminen Flash-soittimessa. Tähän tarkoitukseen taas on jo olemassa useita Drupal-laajennusmoduuleja, erityisesti *Flashvideo*-moduuli, josta itselläni on runsaasti kokemusta. Lopulta todettiin, ettei omaa moduulia kannata tuottaa tähän tarkoitukseen, ja keskityttiin flashvideo-moduulin asetusten konfiguroimiseen samantyyppisen tuloksen saavuttamiseksi. Neo Arenan dokumentointi oli hyödyllinen erityisesti oikeiden *FFMPEG*-asetusten löytämisessä, joskaan järjestelmässä ei päädytty käyttämään samoja konversioparametreja. [29.]

TvExperience käyttää videoiden esittämiseen omaa Flex-pohjaista soitinta, flashvideo-moduuli taas antaa valita muutamasta yleisestä Flash-toistimesta (tai kehottaa vaihtoehtoisesti oman toistimen kustomointiin). Kysyessäni Neo Arenan kehittäjiltä, mikseivät he hyödyntäneet hyvin laajalti käytettyä *JW FLV Player* -videosoitinta, sain vastaukseksi, että se on vain näennäisesti avoimen lähdekoodin sovellus. Sovelluksen ohjelmalisäkkeet toimivat niin, että ne ladataan aina suljetussa muodossa sovelluksen kehittäjien palvelimilta. Tämä taas estää ohjelmalisäkkeiden omat muokkaukset, tai edes lähdekoodiin tutustumisen.

Tästä huolimatta päätettiin ottaa JW FLV Player käyttöön aiempien hyvien kokemusten pohjalta. Sovelluksen lisenssi on kuitenkin halpa, ja se on toiminnoiltaan ja ulkonäöltään helposti muokattavissa sekä myös hyvin tuettu. Sillä on hyvin dokumentoitu ja laaja JavaScript-rajapinta, jonka avulla videotoistimen toiminnot on kätevää integroida muuhun käyttöliittymään.

Neo Arenan alkuperäiseen järjestelmään kuului kaupallinen *Wowza Media Server* Flash-videoiden streamaukseen ja ilmainen *Darwin Streaming Server* samojen videoiden 3GP-versioiden toistamista varten. Molemmat videotyypit käyttävät yhteensopivaa (H.264) MPEG4-formaattia, mutta TvExperiencessä näille kuitenkin tuotetaan erilliset videotiedostot kuvakoon ja laadun optimoimiseksi [29]. Jos

käyttöliittymää ei lasketa, ainoa flashvideomoduulista puuttuva merkittävä toiminto oli mobiililaitteille suunnatun 3GP-version streami. Se jätettiin kokonaan pois, koska mobiiliulottuvuudeksi haluttiin ottaa Sofia Digitalin tuottama mobiili-tv-versio. Potentiaalisen käyttäjäkunnan kannalta se ei välttämättä ollut paras ratkaisu, sillä mobiili-tv on toistaiseksi äärimmäisen harvojen päätelaitteiden käytettävissä, kun 3GP taas on toistettavissa suurella osalla nykypuhelimista. Jopa YouTube käyttää *Darwin Streaming Serveriä ja* 3GP-tiedostomuotoa mobiiliversiossaan. [30.]

Sofia Digital Backstage

Sofia Digital Backstage -järjestelmä on tarkoitettu mobiili-tv-lähetysten tuottamiseen. Siihen kuuluu kaksi palvelinta, joista ensimmäinen on Windows-pohjainen. Sillä hallinnoidaan videotiedostoja ja järjestetään niiden julkaisuaikataulu (EPG) www-pohjaisessa käyttöliittymässä. Järjestelmä tukee laajasti lähes kaikkia videomuotoja, muttei mahdollista varsinaista editointia. Videot on siis tarvittaessa editoitava ennen hallintakoneelle siirtämistä.

Järjestelmään kytketty toinen laite on Linux-pohjainen, ja sen tehtävänä on muuttaa video DVB-H-muotoon ja lähettää signaali eteenpäin. Hämmäntävästi näitä koneita ei ole yhdistetty kuituoptiikalla vaan vanhanaikaisella S-video-kaapelilla. Syynä on ilmeisesti se, että näin voidaan varmistaa, ettei raskaasti pakattu tai muuten korkeatasoinen video saa kumpaakaan konetta jumiutumaan. Käyttövarmuuden parantuminen johtuu siitä, että video on aina valmiiksi samassa yksinkertaisessa muodossa, jonka näytönohjain prosessoi ongelmitta. DVB-H-lähetysten kuvanlaatu on muutenkin sen verran alhainen, ettei S-video-kaapelin signaali huononna kuvanlaatua merkittävästi.

Integraatiomahdollisuudet Sofia Digitalin järjestelmään ovat hyvin rajoittuneet.

Teoriassa hallintopalvelimen www-käyttöliittymää olisi voitu koettaa laajentaa niin, että

se vastaanottaa automaattisesti uudet sivustolle tulleet videotiedostot. Mutta koska kyseessä ei ole avoimen lähdekoodin sovellus, tässä olisi tuskin onnistuttu. Lisäksi todettiin, että www-palvelun videot pitäisi käytännössä suodattaa ja tuottaa parhaat erillisen editointivaiheen kautta. Tämän ja ajan puutteen takia päädyttiin siihen, että toistaiseksi riittää manuaalinen tiedostojen hakeminen Stadi.tv:n www-palvelimelta. Sivuston ylläpitäjien käyttöön oli suunnitteilla toiminto, jolla halutun videotiedoston saisi kopioitua suoraan Drupal-koneelta Sofia Backstagen Windows-palvelimelle (mikäli se ei vaadi editointia), mutta sitä ei ehditty toteuttaa.

3.4 Monikanavaisuus

Www-versio

Www-käyttöliittymän on tarkoitus toimia koko konseptin yhdistävänä tekijänä. Peruskäyttäjien toimintoihin kuuluu sisällön lisääminen, muokkaaminen ja esittäminen. Videotyyppejä ovat omat, itse siirretyt videot ja jo webissä olemassa olevat videot (toistaiseksi tuettuna vain YouTube). Muokkaamiseen voi liittää tekstin lisäksi luokittelu-tageja ja esimerkiksi sijaintitiedon. Yhteisöllisiin työkaluihin kuuluvat muun muassa videoiden kommentointi ja ”pidin/en pitänyt” -äänestysominaisuudet.

Videoista generoituvat automaattisesti tuoreet arvostetuimpien ja selatuimpien videoiden listaukset. Moderaatio-oikeudet omaavat käyttäjät voivat estämisen lisäksi myös nostaa videoita esille näkyvimmille paikoille.

Mahdollisiin myöhemmin toteutettaviin ominaisuuksiin kuuluivat keskustelufoorumien aktivointi, Facebook-integraatio ja käyttäjien omat ryhmät. Eräs kunniahimoisemmista ideoista oli toteuttaa automaattinen suosittelutoiminto, joka osaisi suositella käyttäjälle sisältöjä tämän aiempien valintojen perusteella. Helpoiten tämä olisi toteutettu huomioimalla vain sisältö-tagit. Erityisesti ”pidin/en pitänyt” -äänestyksien

huomioiminen suosittelutoiminnon logiikassa voisi tuottaa jo luotettavia tuloksia. Tällaisille ominaisuuksille olisi varmasti käyttöä, mutta niitä ei kuitenkaan voitu lähteä toteuttamaan, koska tärkeät toiminnot olivat vielä kesken.

Mobiili-versio

Sivustosta luotiin automatisoitu mobiiliversio, joka tuotettiin hyvin nopealla ja kätevällä tekniikalla. Aktivoimalla Drupalin *mobile_theme*- ja *browscap*-laajennusmoduulit, alkaa sivusto tunnistaa mobiililaitteita. Mobiililaitteille näytetään samat sisällöt, mutta *mobile*-ulkoasuteeman läpi, joka suodattaa kaiken ylimääräisen pois käyttöliittymästä. Sivustot latautuvat tällöin siedettävämmällä nopeudella ja järkeväkokoisilla navigointielementeillä.

Browscap-moduuli pitää yllä tietokantaa kaikista tunnetuista mobiililaitteiden lähettäjä-tiedoista, jotka lähetetään jokaisen HTTP-pyynnön yhteydessä. Tietokantaan haetaan automaattisesti tietyin aikavälein päivityksiä, jotta myös tulevat mobiililaitteet voidaan tunnistaa. Tämän tekniikan kustannustehokkuutta on syytä epäillä, koska päätelaitteita on tietokannassa tuhansia ja oletusarvoisen ulkoasun ohittaminen tarkoittaa myös sivuvälimuistin ohittamista, mikä vähentää sivujen käsittelytehoa huomattavasti. Päätelaittekanta kuitenkin tarkastetaan useamman välimuistiportaan kautta, joten sen tietoja ei haeta jokaisella sivunlatauksella.

Mobiiliteemaa pitäisi käytännössä jatkaa ja kustomoida lisää, jotta kuvat ja videotkin olisivat oikeasti käytettävissä mobiililaitteille. Onneksi Drupalin theme override -menetelmä mahdollistaa kaiken näytettävän sisällön muuttamisen toiseen muotoon teemakohtaisesti. Tämä on kuitenkin hyvin työläs prosessi.

Koska mobiiliversiossa ei kuitenkaan käytetä 3GP-videoita, jäänee videoiden www-pohjainen mobiilikäyttö vähäiseksi.

Mobiili-tv-versio

Suunniteltiin, että valitut sisällöntuottajat voisivat luoda suosituista sisällöistä lineaarista videostreamia, joka voidaan näyttää palvelun etusivulla ja mobiili-tv-versiossa.

Käytännössä tähän sisältyisi usea työläs manuaalivaihe, sillä videoiden koostaminen järkevään esityskelpoiseen muotoon vaatisi ihmisen tai vähintäänkin kehittyneen tekoälyn. Tähän vaiheeseen suunniteltiin *Arcada*-ammattikorkeakoulun panosta, mutta tehtävässä voitaisiin käyttää myös muita vapaaehtoisia.

Lineaarisen streamin toteutus jäi valitettavasti suunnitteluvaiheeseen, ja toisaalta sen toteuttamiskelpoisuutta voidaan epäillä. Onnistuminen vaatisi hyvin suurta ja aktiivista käyttäjämäärää syöttämässä uutta sisältöä ja motivoituneita (palkattomia) editoijia. Sisältöjen pitäisi olla myös originaaleja tekijänoikeuskuoppien välttämiseksi. Teknisistä syistä muihin palveluihin ladattuja videoita ei voida kaapata jatkokäsiteltäväksi – ainakaan kovin laillisin keinoin.

Muut ympäristöt

Sisällöille on vielä muita kunnianhimoisia mahdollisia monikanavaisia käyttökohteita. Oudoimpiin kuulunee 3D-virtuaaliympäristö, mutta sitä ei sentään lähdetty toteuttamaan.

Sivuston tuottamat RSS-syötteet on mahdollista hyödyntää esimerkiksi tuomalla sisältöjä muihin palveluihin. *Hubi.fi* on pääkaupunkiseudun karttapalvelu, joka osaa näyttää julkisen liikenteen ja yleiset palvelut reaaliaikaisesti kartalla. StadiTV:n paikkatietoihin kytkettyjen videoiden näkyminen hubi.fi:n kartalla tuottaisi lisäarvoa

molemmille palveluille. RSS-syötteiden avulla tämä olisi teknisesti myös hyvin helppo toteuttaa.

Vielä yksi idea-asteella ollut toistokanava on Arcada-ammattikorkeakoulun tuottama Kaapeli-tv-versio lineaarisesta streamista. Se voisi toimia samalla alkuperäisyydellä ja periaatteella kuin Sofia Digitalin kautta tuotettu mobiili-tv, mutta medioiden erot vaatisivat erillisiä videon käsittelyprosesseja.

3.5 Multimediatiedostojen käsittely

Lisättyjen videotiedostojen suunniteltiin käyvän seuraava prosessi läpi:

- Peruskäyttäjä siirtää videon palvelimelle ja määrittää sille metatagit (kuvaus, tagit, sijainti).
- Videosta luodaan Drupal-järjestelmään ”video”-tyyppinen sivu, mutta se pysyy julkaisemattomana, kunnes *cron-skripti* on ajettu onnistuneesti. Cron suoritetaan automaattisesti tietyin väliajoin (esim. 5 minuutin välein). Isojen videoiden muuntaminen voi kestää lisäksi yhtä kauan tai kauemmin kuin itse videon toisto.
- Videokonversion jälkeen sisältö julkaistaan automaattisesti, minkä jälkeen muut käyttäjät voivat kommentoida ja äänestää sisältöä tai merkitä sen asiattomaksi sisällöksi.
- Sivuston valvojat (engl. *moderator*) näkevät tilastoissa suosituimmat ja kriticoiduimmat sisällöt ja voivat poistaa tai suorittaa nostoja näkyvemmille paikoille. Lisäksi he voivat valita parhaat videot jatkokäsiteltäväksi ja siirtää ne joko suoraan Sofia Digital -järjestelmään tai muualle editoitavaksi.
- Videoiden käsittelyn jälkeen niistä voidaan tuottaa lineaarinen streami, jonka esityskanavana toimii www-palvelu, mobiili-tv, Arcadan kaapeli-tv-versio tai nämä kaikki.

Liite 1 havainnollistaa mediatiedostojen käsittelyprosessin visuaalisesti.

3.6 Pullonkaulat ja resurssien hallinta

Tärkeimmät palvelinoptimoinnit

Videoihin perustuvassa julkaisujärjestelmässä on lukuisia merkittäviä resursseja, jotka voivat ilmaantua hyvinkin nopeasti. Drupalin ja PHP:n osalta on olemassa muutamia erittäin helppoja ja nopeita asetuksia tehokkuuden parantamiseksi. Drupalin suorituskyky-ylläpitosivulta kannattaa valita *taulukon 1* mukaiset asetukset:

Taulukko 1: Yksinkertaisimmat Drupal-optimoinnit.

<i>Asetus</i>	<i>Arvo</i>	<i>Selitys</i>
<i>Sivuvälimuistin tila</i>	<i>normaali</i>	<i>Tämä välimuistiasetus ei tuota häiriöitä, mutta vähentää tietokantakyselyitä.</i>
<i>Sivujen pakkaaminen</i>	<i>käytössä</i>	<i>Käyttäjille lähetettävät tiedot pakataan, eli ne vievät vähemmän kaistaa. Voidaan aktivoida Apachen samantyyppisen pakkaustoiminnon sijaan.</i>
<i>Optimoi CSS-tiedostot</i>	<i>käytössä</i>	<i>Drupalin lukuisat moduulit voivat kaikki tuottaa oman .css-tyylitiedostonsa, jolloin jokainen sivunlataus voi tuottaa kymmeniä ylimääräisiä kyselyitä ja hidastaa sivujen latautumista. Lisäksi vanhemmat IE-versiot eivät tue kuin korkeintaan 31:tä CSS-tiedostoa. Tämä asetus pakkaa kaikki .css-tiedostot yhteen tiedostoon, mikä nopeuttaa sivunlatauksia ja vähentää palvelimen rasitusta.</i>
<i>Optimoi JavaScript-tiedostot</i>	<i>käytössä</i>	<i>Toimii kuten CSS-optimointi, mutta js-tiedostoille.</i>

Drupalin sivuvälimuistin käyttöönottoaminen yksin nopeuttaa kirjautumattomien käyttäjien sivunluomisnopeuksia jopa 500–2 000 %. Nopeus johtuu lähinnä siitä, että tietoa ei haeta tietokannasta vaan nopeasta RAM-välimuistista. Tämä vähentää huomattavasti myös prosessorin ja muistin käyttöä. Kirjautuneet käyttäjät taas eivät huomaa merkittävää nopeutumista pelkän Drupalin välimuistin avulla, koska suurin osa sivusta on kuitenkin luotava dynaamisesti. Muut *taulukko 1*:ssä suositeltavat kolme

asetusta parantavat datansiirtonopeutta huomattavasti kaikille käyttäjille. Tehostumisen aste riippuu paljon sivuston ulkoasun ja sisältömäärän raskaudesta.

Drupalin tuhansista moduuleista löytyy myös ainoastaan suorituskykyyn keskittyviä laajennuksia, kuten *Boost* tai *Throttle*. Myös useat laajat perusmoduulit, kuten *Views*, sisältävät suorituskykyasetuksia. Views mahdollistaa sisältölistauksien välimuistittamisen myös kirjautuneille käyttäjille ja tehostaa kaistankäyttöä tarvittaessa AJAX-tekniikoilla, jolloin sivunlatauksien yhteydessä haetaan ainoastaan muuttuva data.

APC (Alternative PHP Cache)

APC-välimuistin aktivointi tehokkailla asetuksilla vaatii *taulukon 2* mukaiset lisäykset *php.ini*-asetustiedostoon.

Taulukko 2: APC-asetusten optimointi.

Asetus	Arvo	Selitys
<i>extension</i>	<i>apc.so</i>	Aktivoidaan Alternative PHP Cache -laajennus, jonka avulla PHP-tiedostot haetaan välimuistista kovalevyn sijaan.
<i>apc.include_once_override</i>	<i>1</i>	Käytetään skriptitiedostojen dynaamisen sisällyttämiseen optimoitua funktiota PHP:n oletusarvoisen käsittelytavan sijaan.
<i>apc.stat</i>	<i>0 tai 1</i>	0 = ajettavien skriptitiedostojen mahdollisia muutoksia ei tarkisteta erikseen jokaisella sivunlatauksella, vaan vasta Apachen uudelleenkäynnistyksen yhteydessä. Tehokas muutos harvoin muuttuvalla tuotantopalvelimelle, mutta ei kehitysympäristössä. Toimii parhaiten vasta toistaiseksi julkaisemattoman Drupal 7 -version kanssa. [42.]
<i>apc.shm_size</i>	<i>70</i>	Erittäin tärkeä asetus. PHP-tiedostoille palvelimen muistista varattava tila megabitteinä. Jos asetus on liian alhainen, APC:n hyödyt katoavat.
<i>realpath_cache_size</i>	<i>256K</i>	Jos järjestelmässä on huomattava määrä suoritettavia tiedostoja, kannattaa nostaa myös tämän tiedostojen sijaintiin keskittyvän välimuistin kokoa, joka on oletusarvoisesti liian pieni Drupalille.

Oikeilla asetuksilla APC yksin nopeuttaa sivulatauksia jopa 250– 500 % ilman tehollisia haittavaikutuksia. APC:n kaltaisten PHP-kiihdyttäjien on tosin todettu aiemmin joskus johtavan palvelinohjelmiston *segment fault* -virheeseen. Lukemani mukaan aiheuttajana on yhteensopimattomuus ZendOptimizer-optimoijan kanssa, joka kannattaakin poistaa käytöstä. Itse olen omassa käytössä todennut ratkaisun vakaaksi ainakin Linux-palvelimilla, ja raskaille sivustoille myös välttämättömiksi. APC:n korvaaviksi vaihtoehtoiksi voidaan mainita myös *XCache* ja *eAccelerator*, mutta APC on laajimmin käytetty ja tuettu. Usean palvelimen kokonaisuuksissa (klusteri) käytetään usein APC:n sijaan, tai sen rinnalla, *memcached*-tekniikkaa. [31; 32; 33; 34.]

Edellä mainituilla asetuksilla pääsee jo pitkälle: hyvällä optimoinnilla voi yhdellekin palvelimelle keskitetty sivusto käsitellä yli miljoona päivittäistä sivunlatausta [33]. Tätäkin suuremmille käyttäjämäärille skaalautuminen edellyttää lisäpalvelimien lisäksi runsaasti jatko-optimointeja, joiden toteuttamisen laajuus mahdollistaisi useitakin insinööritöitä. [34; 35.]

Levytilan käyttö

StadiTV:n tyyppisen palvelun kaupallinen menestyminen vaatisi erittäin suurta käyttäjämäärää, joille olisi myös tarjottava teknisesti laadukasta videota. Videopalvelutarjontaa riittää, ja esimerkiksi paljon tilaa ja tehoja vaativa teräväpiirto on yleistynyt sekä kuluttujatason kameroissa että videoiden toistolaitteissa. Videoiden laatu vaikuttaa suoraan kaikkiin tehovaatimuksiin, joista huolenaiheena ovat ensisijaisesti levytila ja kaista.

Tiedostoista on säilytettävä originaaliversio, jotta lineaarisen streamin laatu ei kärsisi useista konversioista. Pakkaamaton originaali voi hyvin olla kooltaan useita gigatavuja, jolloin alkuperäisen palvelimen 500 Gb:n kiintolevytila tulee nopeasti vastaan. StadiTV:n arkkitehtuurin osalta tilanne korjattiin toistaiseksi kytkemällä pääasialliseen

palvelimeen *NFS*-palvelin, jonka tehtävänä on tarjota useita teratavuja nopeaa tallennustilaa. *NFS*-levyn yhdistämisen jälkeen se näkyy käyttöjärjestelmässä kuin sisäinen kovalevy. Lisätila saadaan helposti käyttöön luomalla komentorivillä *kuvassa 5* näkyvä *symbolinen linkki* (symlink) Drupalin *files*-kansioon. [36; 37.]

```
ilmari$ sudo ln -s /dev/nfs/files [/drupal-kansio]/default/files
```

Kuva 5. Symbolisen linkin luominen *NFS*-levylle.

3.7 Skaalautumismahdollisuudet

Palvelun teoreettisen käyttäjämäärän nousun mukana nousisivat myös nopeasti palvelimen muistin, prosessorin tehon ja tietokantayhteyden vaatimukset.

Tulevaisuudessa teoreettisesti mahdollinen suurille käyttäjämäärille skaalautuminen vaatisi eri toimintojen jakamista useille palvelimille, esimerkiksi tämälantapaisesti:

- Ensijainen *www*-palvelin, johon asennettu Drupal. Voisi olla myös *palvelinklusteri*, joka käyttää yhteistä *memcache*-välimuistia. Palvelimelle pitäisi asentaa *Varnishin* tapainen käänteinen välityspalvelin staattisten sivujen tehokkaampaa tuottamista varten.
- Erillinen MySQL-palvelin. Usein tosin tietokanta ja *www*-palvelinsovellus toimivat tehokkaimmin samalle palvelimella asennettuna tiedonsiirtoviiveen välttämiseksi.
- Videoiden konversointipalvelin (vaatii paljon muistia ja prosessoritehoa eikä saisi häiritä *www*-palvelinta).
- Ainakin kaksi erillistä videoiden streamauspalvelinta web- ja mobiilistreameille (alustoina Wowza Media Server ja Sofia Digital).
- Erillinen tiedostopalvelin, kuten tämänhetkinen *NFS*-ratkaisu. Mahdollistaa helpon kiintolevytilan lisäämisen ja kaistatehojen jakamisen.

Palvelinarkkitehtuurin lisäksi myös Drupalissa aktiiviset moduulit on testattava tarkkaan muistin ja tietokannan käytön osalta ja päätettävä, mitkä kannattaa poistaa. Useissa isoissa projekteissa valmiiden moduulien sijaan tuotetaan moduulista omaehtoisesti karsittu versio, jotta turhat ominaisuudet eivät syö tehoa. Haittapuolena moduulin päivitettävyys katoaa. [38.]

Erillisten palvelimien yhdistäminen tehokkaalla tavalla olisi jo hyvin vaativa operaatio, ja tekijällä ei ole tällaisesta projektista toistaiseksi kokemusta. [33; 35.]

4 Ulkoisten järjestelmien integrointi Drupal-alustaan

4.1 Alustan soveltuvuus

Drupal on todettu lukuisten esimerkkien avulla erityisen hyvin käyttäjälähtöisten, yhteisöllisten www-palveluiden alustaksi, joka voi myös skaalautua erittäin suurien järjestelmien tarpeisiin. Se on viime vuosina noussut tilastoissa suosituimpien julkaisujärjestelmien kärkeen *Joomla*- ja *WordPress*-järjestelmien kanssa. Nämä järjestelmät on suunnattu enimmäkseen yksinkertaisiin www-projekteihin, kun Drupalia on käytetty eniten teknisesti vaativissa ja laajoissa projekteissa. Viime vuosina Drupal on yleensä saanut julkaisujärjestelmille suunnattujen kilpailujen arvostetuimmat palkinnot. [39; 41.]

Drupal on todettu joustavuutensa ansiosta parhaaksi valinnaksi erityisesti sosiaalisten yhteisöiden rakentamisessa mutta myös videosivustoissa ja muissa suurissa www-projekteissa, jotka vaativat paljon omia muutoksia [14; 23; 39; 40]. Järjestelmää tuskin on suunnattu alun perin erityisesti tähän suuntaan, mutta käyttäjien kehittelemät laajennusmoduulit luonnollisesti noudattavat yleisien www-trendien kehitystä. Videon ja sosiaalisen median lisäksi lähes kaikille www-sivustoille yleisesti halutuille

toiminnoille on valmis, ilmainen moduuli, jonka voi ottaa käyttöön ilman ylimääräistä ohjelmointia. Tai jos on muutosten tarve ja taidot riittävät, voi Drupalin omassa yhteisössä aktivoitua ja ehdottaa muutoksia koodiin tai ominaisuuksiin.

Joskus käy niin, että saman sivuston kaksi eri moduulia eivät ole yhteensopivia toistensa kanssa, mikä johtuu yleensä siitä, että molemmat toteuttavat saman laajennusominaisuuden, joskin hieman eri tavalla. Näistä moduuleista usein toinen sitten poistuu käytöstä, ja sen kehittäjät saattavat yhdistää voimansa kilpailevan moduulin tekijöiden kanssa. Näin tapahtuu varsin usein, ja duplikaattimoduulien määrä tuottaa joskus myös ylimääräistä selvitystyötä. Esimerkiksi *cckasetracker*-moduulin projektisivulla suositellaan siirtymään *comment_cck*-moduulin käyttöön, ja *comment_cck*:n sivuilla taas ilmoitetaan sen vanhentumisesta ja suositellaan siirtymistä samantapaisen *comment_driven* käyttöön.

Uskon Drupalin suosion liittyvän sen ohjelmoijaystävälliseen helposti laajennettavaan rakenteeseen, jonka avulla alustaa voi käyttää periaatteessa mihin tahansa www-tekniikkaan. Omien kokemuksieni mukaan monet käyttäjät kokevat järjestelmän käytön oppimiskäyrän kilpailevia järjestelmiä jyrkemmäksi. Omasta mielestäni sivuston kehittäjän tehtäviin kuuluu käyttöliittymän muokkaaminen käyttäjien tarpeisiin sopivaksi. Usein hämmennys johtuu liiallisista toiminnallisuuksista, ja ne voidaan piilottaa esimerkiksi käyttöoikeuksia rajoittamalla. Käyttöliittymän helppokäyttöisyys on kuitenkin parantunut huomattavasti jokaisessa versiossa. Piakkoin julkaistavan Drupal 7:n kehityksessä on käytettävyyden parantaminen ollut yhtenä erityisenä painopisteenä. [42.]

Avoim järjestelmä

Drupal-yhteisö kannustaa automaattisesti osallistumaan moduulikehitykseen, käyttöliittymän dokumentoimiseen ja kääntämiseen. Tämä onkin lähes aina vapaaehtoistyötä, josta saavutettu hyöty ei useimmiten ole suoraan nähtävissä. Avoimen

lähdekoodin yhteisöissä aktiivisuudesta saa myös jonkinlaista uskottavuutta. Jos osoittaa olevansa hyödyksi, ovat myös muut yleensä halukkaampia auttamaan.

Oman koodin jakaminen muille ei ole mikään välttämättömyys. Jos kehittämästään moduulista uskoo saavansa elintärkeää kilpailuetua, voi sen pitää suljettuna. Tällöin ei kuitenkaan pääse hyötymään laajemman yleisön hyödyistä, kuten testikäyttäjistä ja moduulin kehitykseen osallistuvista samoista toiminnoista kiinnostuneista ohjelmoijista. Ulkoisten tahojen työpanos voi pitää moduulin ajantasaisena ja tietoturvallisena.

Avoimen lähdekoodin periaatteet siis riippuvat laajoista yhteisöllisistä toiminnoista ja osaamisen jakamisesta kaikkien hyväksi. Tämän ansiosta Drupal ja moni muu avoimen lähdekoodin ohjelmisto myös kukoistaa ja pystyy venymään lähes kaikilla osa-alueilla parempiin tuloksiin kuin suljetut kilpailijat. Sovelluksen ilmaisuus ei tarkoita, etteikö sen avulla voisi tehdä liiketoimintaa: Drupalia käytetäänkin erittäin runsaasti kaupallisissa projekteissa. [40.]

4.2 Avoimen lähdekoodin työkalut

LAMPP-arkkitehtuuri

LAMPP on yhdistelmä sanoista **L**inux, **A**pache, **M**ySQL, **P**HP (ja **P**erl tai **P**ython), joka on nykyään suosituin yhdistelmä avoimen lähdekoodin www-ympäristöissä. Linux-käyttöjärjestelmä ei ole välttämättömyys, samat työkalut löytyvät myös Windowsille ja Mac OS:lle. Itse käytän aktiivisesti tätä kehitysympäristöä kaikkien mainittujen järjestelmien kanssa. Käytännössä oikeat tuotantopalvelimet ovat kuitenkin aina Linux-koneita, kun Windows ja Mac OS toimivat itselläni useimmiten kehitysympäristöinä.

Sovellukset voi asentaa erikseen, mutta kaikille alustoille löytyy useita toimivia pakettiratkaisuja, esimerkiksi XAMPP, WAMPP, MAMP, joissa kaikissa

oletusasetukset on tarkoitettu kehitysympäristöön. Lisäksi on ammattimaisempaan käyttöön tarkoitettu Zend Server, josta on myös maksullinen versio.

Kaikille osa-alueille löytyy vaihtoehtoisia tekniikoita, mutta suosituimmat sovellukset tuntuvat pysyvän suosittuina kerran siihen asemaan päästyään. Kriittinen massa avoimen lähdekoodin projekteissa tarkoittaakin lähes automaattisesti sovelluksen menestystä, ainakin niin kauan kuin alkuperäisen sovelluksen tekijät eivät itse estä kehitystä. Vaikka he estäisivätkin, syntyy yleensä käyttäjä-kehittäjien toimesta uusi vaihtoehtoinen sovellus eli variantti vanhan lähdekoodin pohjalta, joka voi lähteä täysin uusille urille ja ohittaa alkuperäisversion suosiossa.

Esimerkiksi erittäin suositun MySQL:n kehittyminen vaikuttaa tällä hetkellä kyseenalaiselta, koska *Sun Microsystems* osti sen hiljattain. Sun taas päätyi tietokantajätti *Oraclen* ostamaksi. Lähdekoodin avoimuuden ansiosta MySQL:lle on kuitenkin olemassa jo sen alkuperäisten kehittäjien luoma ja sen kanssa yhteensopiva, varmasti avoimena pysyvä MariaDB. [43; 44.]

Linuxille on syntynyt runsaasti variantteja heti sen kehityksen alusta lähtien, ja palvelinkäyttöönkin on tarjolla lukuisia ilmaisia ja kaupallisia ratkaisuja – ja näin on varmasti tulevaisuudessakin.

Apachen httpd-www-palvelinytimeistä on olemassa useita erikoistuneita ratkaisuja, kuten aiemmin mainitut lighttpd ja nGinx, jotka on tarkoitettu erityisesti valtavien sivunlatausmäärien hallintaan. Apache itsessään on erilaisten laajennuksien ansiosta hämmästyttävän monipuolinen, mutta samalla näiden laajunnusten määrä syö tehoa. Tämän takia monet suositut palvelut ovat joutuneet keksimään keinoja, joilla ne pystyvät hyödyntämään useamman eri palvelinsovelluksen ominaisuuksia.

Esimerkiksi *YouTube*, joka perustuu muokattuun LAMPP-arkkitehtuuriin, on jakanut toimintojaan niin että saman sivuston staattiset sisällöt tuotetaan `lighttpd`:llä, kun Apache hoitaa raskaan tiedonkäsittelyn. [45.]

PHP taas on ohjelmointikielenä tuottanut vähemmän variantteja. Sille on kuitenkin olemassa suuri joukko ilmaisia laajennuksia, kuten *PEAR*-kirjastot, jotka mahdollistavat suuremman ominaisuuskirjon. Ominaisuuksien lisääminen kuitenkin vaikuttaa myös järjestelmän raskauteen, minkä takia on tärkeää, että kaikki laajennukset eivät ole aina automaattisesti päällä, vaan käytetään vain sitä, mitä tarvitaan. PHP:lle on lisäksi myös olemassa oikeita variantteja, esimerkiksi Facebookin tuottama *HipHop for PHP*, joka muuttaa koko kielen kääntämistapaa. [46.]

Videotiedostojen muuntaminen

Kaikki StadiTV-projektin audio- ja videokäsittely hoidetaan avoimella `FFMPEG`-ohjelmalla, joka tukee laajennuksiensa avulla lähes kaikkia tunnettuja multimediaformaatteja. Alusta toimii lisäksi Windowsissa, Mac OS:ssa ja useimmissa Linux-varianteissa.

`FFmpeg`iä käytetään komentoriviltä, ja koska valittavissa olevat parametrit ovat hämmästyttävän laajoja, on peruskäytönkin opettelu varsin työlästä. Koska käyttäjien tuottamat videot saattavat olla mitä tahansa formaattia, pitäisi käyttää parametrejä, jotka toimivat kaikille formaateille. Lopputulos on aina jonkinasteinen kompromissi laadun, tehokkuuden ja tiedostokoon kesken.

Optimaalisten komentojen löytäminen ei ole mitenkään itsestäänselvää. Sama käsky voi toimia yhdelle tiedostolle mutta saattaa jumittua tai tuottaa korruptoituneen käännöksen toisen tiedoston kanssa. Tätä voitaisiin välttää muokkaamalla kullekin formaatille omat parametrit, mutta Drupalin flashvideo-moduuli ei tue tätä. Olisi kyllä mahdollista

toteuttaa flashvideota parantava laajennusmoduuli, mutta sitä ei StadiTV-projektissa tehty.

Oikeiden säätöjen löytäminen vaati paljon kokeiluja yritys ja erehdys -menetelmällä. Debug-tiedot eivät aina ole kovin avuliaita. Törmäsin jopa siihen ongelmaan, että satoja merkkejä pitkä komentorivi ei tallentunut oikein Drupalin ylläpitosivuilla. Tämä johtui siitä, että kentällä oli 256 merkin rajoitus, jota ei huomaa komentoriviä kopioitaessa. Tämä korjattiin itsekehitetyn moduulin avulla, ja ratkaisu toimi ongelmitta. Turhan koodin välttämiseksi ehdotin myös korjausta *flashvideo*-moduulin kehittäjälle, joka ei kuitenkaan lähtenyt toteuttamaan muutosta vedoten ongelman harvinaisuuteen ja kiertämismahdollisuuteen.

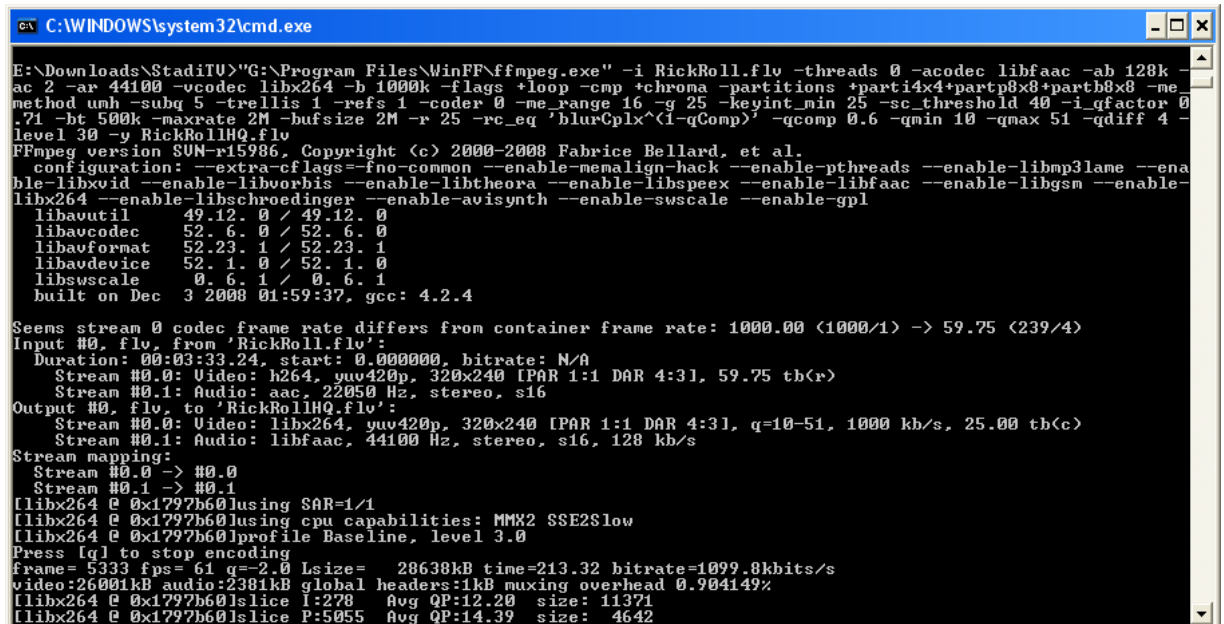
Lisähaasteena sekä mediaformaattit että FFMPEG-päivittyvät jatkuvasti, ja uusissa versioissa voi ilmaantua yhteensopivuusongelmia tai muita sovellusvirheitä. Päivittymistä ei välttämättä uskalla tehdä kokeilematta ensin uusimman version toimintaa, koska videokäännösten toimimattomuus tuottaisi ongelmia. Testaaminen kaikilla videoformaateilla erikseen aina päivityksen yhteydessä olisi kuitenkin erittäin työlästä. Olisi mahdollista tehdä skripti, joka hoitaa testaukset automaattisesti. Siitäkin olisi mahdollista tehdä moduuli.

StadiTV:n flashvideo-asennuksen FFMPEG-videokonversiokäsky määrittyi lopulta seuraavanlaiseksi:

```
-i @input -threads 0 -acodec libfaac -ab 128k -ac 2 -ar 44100 -vcodec libx264 -b 1000k -flags +loop -
cmp +chroma -partitions +parti4x4+partp8x8+partb8x8 -me_method umh -subq 5 -trellis 1 -refs 1 -
coder 0 -me_range 16 -g 25 -keyint_min 25 -sc_threshold 40 -i_qfactor 0.71 -bt 500k -maxrate 2M -
bufsize 2M -r 25 -rc_eq 'blurCplx^(1-qComp)' -qcomp 0.6 -qmin 10 -qmax 51 -qdiff 4 -level 30 -y
@output
```

Kuvassa 6 hyödynnetään parametrejä ja muunnetaan RickRoll.flv-tiedosto H.264-videokoodekin (libx264) ja AAC-äänikoodekin (libfaac) käyttäjäksi. Tärkeimpiin

parametreihin kuuluvat `-acodec libfaac`, `-ab 128k`, `-vcodec libx264` ja `-b 1000k`, jotka määrittävät käytetyt koodekit ja kaistanleveyden (bitrate).



```
C:\WINDOWS\system32\cmd.exe

E:\Downloads\StadiTV>"G:\Program Files\MinFF\ffmpeg.exe" -i RickRoll.flv -threads 0 -acodec libfaac -ab 128k -
ac 2 -ar 44100 -vcodec libx264 -b 1000k -flags +loop -cmp +chroma -partitions +parti4x4+partp8x8+partb8x8 -me
method umh -subq 5 -trellis 1 -refs 1 -coder 0 -me_range 16 -g 25 -keyint_min 25 -sc_threshold 40 -i_qfactor 0
.71 -bt 500k -maxrate 2M -bufsize 2M -r 25 -rc_eq 'blurCplx^(1-qComp)' -qcomp 0.6 -qmin 10 -qmax 51 -qdiff 4 -
level 30 -y RickRollHQ.flv
FFmpeg version SUN-r15986. Copyright (c) 2000-2008 Fabrice Bellard, et al.
configuration: --extra-cflags=-fno-common --enable-memalign-hack --enable-pthreads --enable-libmp3lame --ena
ble-libxvid --enable-libvorbis --enable-libtheora --enable-lspspeex --enable-libfaac --enable-libgsm --enable-
libx264 --enable-libschröedinger --enable-avisynth --enable-swscale --enable-gpl
libavutil 49.12.0 / 49.12.0
libavcodec 52.6.0 / 52.6.0
libavformat 52.23.1 / 52.23.1
libavdevice 52.1.0 / 52.1.0
libswscale 0.6.1 / 0.6.1
built on Dec 3 2008 01:59:37, gcc: 4.2.4

Seems stream 0 codec frame rate differs from container frame rate: 1000.00 (1000/1) -> 59.75 (239/4)
Input #0, flv, from 'RickRoll.flv':
  Duration: 00:03:33.24, start: 0.000000, bitrate: N/A
    Stream #0.0: Video: h264, yuv420p, 320x240 [PAR 1:1 DAR 4:3], 59.75 tb(r)
    Stream #0.1: Audio: aac, 22050 Hz, stereo, s16
Output #0, flv, to 'RickRollHQ.flv':
  Stream #0.0: Video: libx264, yuv420p, 320x240 [PAR 1:1 DAR 4:3], q=10-51, 1000 kb/s, 25.00 tb(c)
  Stream #0.1: Audio: libfaac, 44100 Hz, stereo, s16, 128 kb/s
Stream mapping:
  Stream #0.0 -> #0.0
  Stream #0.1 -> #0.1
[libx264 @ 0x1797b60]using SAR=1/1
[libx264 @ 0x1797b60]using cpu capabilities: MMX2 SSE2Slow
[libx264 @ 0x1797b60]profile Baseline, level 3.0
Press [q] to stop encoding
frame= 5333 fps= 61 q=-2.0 Lsize= 28638kB time=213.32 bitrate=1099.8kbts/s
video:26001kB audio:2381kB global headers:1kB muxing overhead 0.904149%
[libx264 @ 0x1797b60]slice I:278 Avg QP:12.20 size: 11371
[libx264 @ 0x1797b60]slice P:5055 Avg QP:14.39 size: 4642
```

Kuva 6. Ffmpeg-videokonversion testaus Windowsin komentorivillä.

Käyttämäni komentorivi sisältää todennäköisesti myös epäoleellisia parametrejä, joiden vaikutuksia kokeilin lukuisissa testikonversioissa. Sain neuvoja Arcada-ammattikorkeakoulun Niclas Hallgreniltä ja FFmpeg-projektin dokumentaatiosta. [47.]

4.3 Kaupalliset työkalut

Avoim mutta kaupallinen käyttöjärjestelmä

StadiTV-projektin pääpalvelimen käyttöjärjestelmänä toimii maksullinen Red Hat Enterprise Linux. Valintaan vaikutti ehkä eniten se, että Elisan kaltainen suuryritys mieluiten maksaa palvelusta, mikäli sillä saavutetaan hyvä takuu ja tuotetuki

ongelmatilanteiden ratkaisuun. Kaupallisesta palvelimesta voi olla myös hieman haittaa: koska käyttöjärjestelmä ei ole yhtä laajalti käytössä kuin ilmaiset vaihtoehdot, siihen on tarjolla yleensä hieman vähemmän suoraan toimivia ohjelmia.

Tämän ei kuitenkaan ole varsinainen este, koska Unix-maailmassa pystytään yleensä lähdekoodista kääntämällä ottamaan sama sovellus käyttöön lähes kaikilla alustoilla. Ongelmatilanteitakin esiintyy, mutta esimerkiksi useimmat Debian-ohjelmat toimivat suoraan RHEL-alustalla. Käytössä oleva Linux-muunnelma on kuitenkin aina otettava huomioon www-palveluja suunniteltaessa.

Videoiden lähetys ja toisto

Wowza Media Streaming on kaupallinen Flashin videostreamin (RTMP) lähetysohjelmisto. Sille on olemassa myös ilmaisia avoimen lähdekoodin versioita, kuten *Red5*. Tämä oli kuitenkin ainakin palvelun kehittämisen ajankohtana liian raskas ja epäkäytännöllinen. Lisäksi Wowza on NeoArenan kehittäjien suosittelu. [21; 29.]

Sovellukselle ei hankittu 995:en dollarin hintaista kaupallista lisenssiä, koska ilmaisen version rajoitukset riittivät testikäyttöön ja koska sovelluksen integraatio ei ollut välttämätöntä palvelun toiminnan kannalta.

Videoiden toistossa käytetty *Longtail Player* -videosoitin tunnetaan myös nimellä *JW FLV Media Player*. Soitin mainostaa olevansa Internetin suosituin avoin Flash-soitin, mikä on todennäköisesti totta. Itse olen käyttänyt sitä lukuisissa www-projekteissa neljän vuoden ajan. Ainakin sen lähdekoodi on avoin, ja soitin on erittäin monipuolinen.

Toistimeen on olemassa lukuisia ohjelmalisäkkeitä, sisäänrakennettu mainosjärjestelmä, ulkoasun kätevä muuttaminen ulkoasuteemoilla (engl. *skin*) ja erityisen tärkeä

JavaScriptiä tukeva API. Näiden avulla soittimen integrointi muuhun sivustoon onnistuu kohtuullisella vaivalla.

Koska StadiTV on kaupallinen palvelu, piti sovellukselle hankkia 59 € maksanut lisenssi.

4.4 Modulaarinen rakenne

Drupal-alustan parhaisiin puoliin kuuluu, että kaikkien moduulien ominaisuuksia on helppo laajentaa ja jatkokehittää omilla moduuleillaan koskematta muiden koodiin. Myös olemassa olevia moduuleita voi muokata suoraan, mutta se vaikeuttaa päivityksiä, koska muutokset olisi tehtävä käsin uudelleen aina moduulipäivityksen jälkeen.

Koska järjestelmä on alusta asti tehty modulaariseksi, on myös kiinnitetty huomiota siihen, miten jopa tuhannet eri moduulit saadaan toimimaan samassa järjestelmässä. Tämä onnistuu yleensä erilaisilla hook-funktioilla ja Drupal API:n yleisillä toiminnoilla, joilla luodaan esimerkiksi lomakkeita, valikoita, käännöksiä tai ulkoasuun vaikuttavia muutoksia.

Tärkeimmät yleiset moduulit

Kaikki moduulit pääsevät kiinni toistensa ominaisuuksiin Drupal API:n avulla ja voivat toimia sen avulla yhteistyössä häiritsemättä toisiaan. Drupalin tuhansista moduuleista suurin osa onkin integroitu toisiinsa jollakin tasolla. Esimerkiksi *views*-moduulia ilmoittaa tukevansa suoraan yli 300 eri moduulia ja *content*-moduulia yli 450. Tukeminen ei tarkoita vain yhteensopivuutta vaan nimenomaan toisen moduulin ominaisuuksien laajentamista.

StadiTV-projektissa päärooleissa olivat yhteisötoimintoja parantavat moduulit, kuten *comment*, *votingapi*, *vote_up_down*, ja multimodiamoduulit, kuten *flashvideo* ja *emfield*. Nämä olivat vain alkuvaiheessa aktivoidut moduulit. Yhteisöllisyystoimintojen laajentamiseksi voitaisiin käyttää myös lukuisia muita moduuleja, kuten *forum*, *advanced_forum*, *user_relationships*, *friendlist*, *phpfreechat*, *og*, *privatemsg* ja *userpoints*.

Liite 2 sisältää luettelon kaikista aktivoiduista moduuleista.

Projektia varten kehitetyt moduulit

StadiTV-projektia varten kehitettiin uusi moduuli, jonka pääasiallinen käyttötarkoitus oli parantaa muutamia flashvideo-moduulin asetuksia.

Kuva 7 näyttää, miten omalla moduulilla voidaan Drupalin Form API -rajapinnan avulla muokata toisen moduulin asetuksia hyvin pienellä vaivalla.

```

69  /*
70  Hook form_alter
71  */
72  function stadtiv_form_alter(&$form, $form_state, $form_id){
73      // arg() palauttaa sivun argumenttityypin
74      $node_type = arg(4);
75
76      // Muutetaan FFMPEG merkkirajoitusta, nostettava se yli 256! Vaikka yli tuhanteen. Textareassa on myös helpompi editoida.
77      if(isset($form['ffmpeg']) && $form_id == 'flashvideo_settings_form' && $node_type){
78          $form['ffmpeg']['flashvideo_', $node_type]['_cmd']['#maxlength'] = 1500;
79          $form['ffmpeg']['flashvideo_', $node_type]['_cmd']['#type'] = 'textarea';
80          $form['ffmpeg']['flashvideo_', $node_type]['_cmd']['#rows'] = 6;
81      }
82  }
83

```

Kuva 7. Stadtiv.module-tiedoston lähdekoodia.

4.5 Sosiaalisten mediasivustojen rajapintojen hyödyntäminen

Lähes kaikilla merkittävillä yhteisöllisillä www-palveluilla on jonkinlainen avoin rajapinta, jota muut sivustot voivat hyödyntää. Sellainen on esimerkiksi YouTubessa (video-embed), Facebookissa (Facebook connect) ja GoogleMapsissa (Gmaps Integration).

Rajapintojen implementointi saattaa usein olla työlästä, mutta niille kaikille on valmiita Drupal-moduuleja: *emfield* (*embedded media field*, joka tukee YouTuben lisäksi muitakin videopalveluja), *fbconnect* ja *gmap*. StadiTV-projektissa oli käytössä *emfield* ja *gmap*. *Fbconnect* olisi todennäköisesti aktivoitu myöhemmin, ja se olisi mahdollistanut ainakin StadiTV:n videosisältöjen muuttamisen Facebookin linkeiksi. Sen avulla videoiden suosittelu katsojen sosiaalisille kontakteille yksinkertaistuisi ja voisi tuoda lisää käyttäjiä.

Myös monet pienet palvelut ylläpitävät avoimia rajapintoja. Esimerkiksi suomalainen videojulkaisuun erikoistunut *Floobs* halusi osallistua projektiin. Kävin Floobsin toimistossa Töölössä tutustumassa sen järjestelmiin. Siellä oli kehitysasteella Flash-välitteinen webkamera-videoiden suoratoisto ja tallennus sekä myös itsekehitetty Drupal-moduuli, joka mahdollistaa videoiden lisäämisen Drupal-sivustoilta suoraan sen järjestelmään. Tästä ei kuitenkaan olisi ollut juuri hyötyä StadiTV:n kannalta, koska muualle lisättyjä videoita ei voitaisi jatkokäsittellä esimerkiksi mobiili- ja kaapelistreamin käyttöön. Näin ollen Floobs-integraatiota ei lähdetty toteuttamaan sen pitemmälle. Valitettavasti Floobs myös ajautui konkurssiin vuoden 2010 alussa. [48.]

Muita epätodennäköisiä mutta mahdollisia hyödynnettäviä palveluita olivat ainakin *qik.com* ja luvussa 3.4 esitelty *hubi.fi*.

4.6 Käyttäjätasot ja käyttöoikeudet

Käyttäjäroolit

Drupal mahdollistaa helposti rakennettavan mutta erittäin monipuolisen käyttöoikeusmatriisin. Kaikki moduulit voivat määrittää rajattomasti eri käyttöoikeuksia, ja ylläpitäjä voi luoda rajattomasti rooleja oletusarvoisten *kirjautunut käyttäjä*- ja *anonyymi käyttäjä* -roolien lisäksi.

Kullekin roolille valitaan halutut käyttöoikeudet whitelist-menetelmällä. Jos käyttäjällä on useita rooleja, muodostetaan laajemmat käyttöoikeudet laskemalla sallitut oikeudet yhteen. Projektin tarpeisiin luotiin kolme uutta käyttäjätasoa: *sisällöntuottaja*, *moderaattori*, *ylläpitäjä*. Kuva 8 havainnollistaa käyttöoikeuksien hallintalogiikkaa.

Käyttöoikeus	anonyymi käyttäjä	sisäänkirjautunut käyttäjä	moderaattori	sisällöntuottaja	ylläpitäjä
moduuli admin_menu					
access administration menu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
display drupal links	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
moduuli advanced_help					
view advanced help index	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
view advanced help popup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Kuva 8: Osa StadiTV:n roolien käyttöoikeusmatriisista.

Sisällön valvonta ja esilletuominen

Sisällön moderointiin suunniteltiin passiivista käyttäjälähtöistä valitusmenetelmää. Jos käyttäjä löytää loukkaavaa tai laitonta materiaalia, hän voi painaa ”merkitse asiattomaksi sisällöksi”-linkkiä. Tämän jälkeen sivu ilmestyy moderaattori-roolin omaavien käyttäjien nähtäväksi, ja he voivat poistaa sisällön palvelusta.

Eräissä palveluissa koko moderointi on luovutettu perustason käyttäjille. Tarpeeksi suuri valitusmäärä johtaa automaattiseen sisällön poistoon ilman erillistä moderointivaihetta. Tämä on käytössä esimerkiksi ohjelmoijille suunnatussa *stackoverflow.com:ssa*.

Käyttäjät voivat valittamisen sijaan myös suositella videota napin painalluksella. Tämän jälkeen moderaattori- tai sisällöntuottaja-roolin omaavat henkilöt näkevät suositellut sisällöt ja voivat nostaa ne esimerkiksi etusivulla näkyvämpään paikkaan. Tämäkin voidaan automatisoida poistamalla tarkistusvaihe. Sisällöntuottajilla on erikseen myös mahdollisuus kopioida suosittuja sisältöjä jatkokäsiteltäväksi mobiili- tai kaapelistreamia varten. Näin suunniteltiin, muttei ehditty toteuttaa.

5 StadiTV-palvelun testaus

5.1 Pilotti: videoiden lähetys mobiililaitteilla

Testiasetelma

Kesäkuussa 2009 toteutettiin palvelun sisäinen pilotointi. Kehittäjäryhmä lähti kaupungille ja otti käyttöönsä seuraavat mobiililaitteet:

- Sony HDR-CX11E, digitaalivideokamera
- Canon PowerShot SX1, kuva- ja videohybridikamera (HD)
- i-Buddie, minilaptop, jossa on 3G-mobiililaajakaista
- useita Nokian kamerapuhelimia (esim. N95).

Ryhmän jäsenten tehtävänä oli kokeilla laitteita käytännön tilanteissa ja pyrkiä lähettämään videot palveluun suoraan paikan päältä. Digitaalivideokamera ja

hybridikamera eivät luonnollisesti kykene lähettämään videoita suoraan palveluun, joten niiden muistikortit oli ensin purettava i-Buddie-minilaptop. Tämä kannettava tietokone käytti SIM-korttiin perustuvaa 3G-yhteyttä stadt.vi-palvelimelle yhdistämiseen. Yhteys oli myös mahdollista ottaa hyödyntämällä kaupungin avoimia verkkoja, jolloin tiedonsiirtonopeus on huomattavasti nopeampaa.

Matkapuhelimen lisäksi videoita lähetettiin testaajien henkilökohtaisilla kamerapuhelimilla, jotka siirrettiin palveluun puhelimen dataliittymän ja www-selaimen avulla.

Tällä testiasetelmalla pyrittiin testaamaan

- mobiilirajapinnan käyttöliittymää eri laitteilla
- laitteiden käyttökelpoisuutta mobiilikäyttöön tosielämässä
- mahdollisia vaikeuksia tiedonsiirrossa
- erimuotoisten videoiden automaattista MP4/H.264-konversiota.

Käyttäjät lähettivät odotetusti muutamia kymmeniä videoita, jotka ovat edelleen nähtävissä palvelussa. Videoiden laatu vaihteli käytössä olleiden kameralaitteiden mukaan.

Testin todettiin onnistuneen kohtuullisen hyvin, seuraavin huomioin ja ratkaisuin:

1. Mobiililiittymien datayhteys saattaa katketa kesken ison tiedonsiirron. Käyttäjä ei saa tästä välttämättä mitään ilmoitusta, mikä voi aiheuttaa sekaannusta.
→ *Ei korjattavissa palvelinpäässä, tästä voidaan tehdä varoitus käyttäjälle.*
2. Harva käyttäjä haluaa lähettää videoita kannettavalta suoraan paikan päältä. Paikan päällä otettu video on kätevintä lähettää palveluun kotoa.
→ *Tämä on käyttäjän valittavissa. Tulevaisuudessa päätelaitteet ja datayhteydet oletettavasti*

kehittyvät käyttökelpoisemmiksi.

3. Optimoimaton www-sivusto on jokseenkin hidas ja raskas puhelimen selaimelle.

→ Sivustolla otettiin käyttöön browscap-, mimedetect- ja mobile_theme -moduulit. Ne tunnistavat mobiililaitteiden käyttäjät ja esittävät selaimelle automaattisesti mobiilioptimoidun ulkoasun.

4. Yhden videotyyppin muuntaminen epäonnistui: Sonyn .MTS-muotoinen video.

→ MTS saatiin myöhemmin muuntumaan uusilla FFMPEG-asetuksilla, jotka pakottivat sille tietyn kuvantoistonopeuden (framerate). Muutosta ei kuitenkaan ole järkevää toteuttaa kaikille muille videotyypeille, mutta flashvideo ei tue erillistä käsittelytapaa muille järjestelmille.

5.2 Käyttöliittymän kehitys ja testaus

Palvelun käyttöliittymän suunnittelussa ja toteutuksessa sain apua Forum Viriumin Kimmo Tukiaiselta. Hän suunnitteli palvelun ulkoasun uusiksi useaan kertaan käyttöliittymän parantamiseksi, jolloin sain itse keskittyä tekniseen toteutukseen. Me molemmat olimme kuitenkin erittäin työllistettyjä myös muiden töiden kanssa, joten ulkoasun viimeistely jäi jälkeen kun ominaisuudet lisääntyivät jatkuvasti. Sijaintitietojen lisääminen ja karttaominaisuudet mullistivat käyttöliittymän kokonaan. Valitettavasti alkuvaiheen yksinkertaista mutta toimivaa ulkoasua ei voitu enää hyödyntää karttatoimintojen kanssa.

Projektin pieni budjetti ei mahdollistanut ulkopuolisen graafikon saamista kuin vasta aivan lopussa, noin kaksi viikkoa ennen työnantajani vaihtumista. Graafikolla oli aiempaan verrattuna hyvin erilaisia näkemyksiä – joskin hyvin osuvia.

Käyttöliittymän tärkeimpiin puoliin kuuluu perustoimivuus kaikilla käyttäjätyleypeille. Erityisesti huononäköiset pitää huomioida suunnitteleamalla sivun tekstiosat niin, että niitä voi skaalata vapaasti. Värimaailma taas ei saa huonontaa värisokeiden käyttökokemusta. [49, s. 169–171.]

Käytännössä loppukiire johti kuitenkin siihen, että sekä ulkoasu että toiminnallisuudet jäivät keskeneräiseen vaiheeseen työpaikkani vaihtuessa. Käyttöliittymän epäonnistuminen on varma tae palvelun epäonnistumiselle, vaikka tekninen puoli toimisikin. Näin ollen palvelun laajempi testaaminen ei voi edetä ennen ulkoasun korjaamista.

Käyttöliittymän testaus jäi käytännössä siis siihen, mitä palautetta saatiin projektin suunnittelutilaisuuksissa. Jos aika ja resurssit olisivat antaneet myöten, olisi varmasti ollut aiheellista pitää käyttöliittymälle ylimääräinen palautekierros koko ryhmän toimesta. Tämän toteutuksessa olisi auttanut, jos testattavat ominaisuudet olisivat ehtineet toteutua aiemmin.

Palvelun ulkoasun muutokset ovat nähtävissä *liitteessä 3*.

5.3 Ohjelmoinnin laadunvalvonta

Yksi projektin merkittävimmistä puutteista oli järjestelmällisen ohjelmoinnin laadunvalvonta. Olin ainoa ohjelmoija, ja samalla myös vastuussa koko teknisestä puolesta. Yksikössä ei ollut lisäkseni muita, jotka olisivat pystyneet arvioimaan tuotetun koodin mahdollisia ongelmakohtia.

Käytännössä teknisen laadun seuraaminen olisi pitänyt ulottaa Drupal-sivuston ylläpitoon, sillä tuottamani koodin määrä keskittyi loppujen lopuksi lähinnä yhteen moduuliin ja teemaan. Sinänsä itse pidin yksilöllisestä vastuutilanteesta, ja olen siihen myös tottunut, mutta ohjelmoinnin laatu voi kärsiä valvonnan puutteesta.

Insinööriyöprojektini jälkeen ehdin uudessa työpaikassani tottua *Subversion*-nimiseen versionhallintaohjelmaan ja *OpenAtrium*-projektihallintajärjestelmään (joka on Drupal-

pohjainen). Molemmat ovat erittäin hyödyllisiä työkaluja, jotka olisi kannattanut ottaa mukaan jo StadiTV-projektiin.

5.4 Tietoturvariskien hallitseminen

Www-palvelun mahdolliset tietoturvariskit ovat Drupal-alustan ansiosta huomattavasti pienemmät kuin kokonaan itseohjelmoidussa järjestelmässä (riippuen tietysti ohjelmoijasta). Drupalin ytimeen ja laajennusmoduuleille tulee jatkuvasti tietoturvapäivityksiä, mikä ei kuitenkaan johdu aukkojen runsaudesta vaan käyttäjäyhteisön aktiivisuudesta. Moduulipäivityksistä ei ole kuitenkaan hyötyä, ellei niitä ota säännöllisesti käyttöön.

Ylläpitäjä voi väärillä asetuksilla aiheuttaa sivustolle haavoittuvuuksia, mutta ne ovat ensisijaisesti vältettävissä järkevillä syöttömuotoasetuksilla. Erityisesti PHP-suodin on vaarallinen ja väärissä käsissä mahdollistaa tietojen varastamisen ja koko palvelimen sekoittamisen. Rajoitettua HTML-suodatinta on käytettävä kaikissa käyttäjän määrittämissä sisällöissä XSS-haavoittuvuuksien estämiseksi.

Drupalin käyttäjätunnusten tietokanta on MD5-salattu, mutta ilman niin sanottua *suolaa*, mikä teoriassa parantaa tunkeutujan mahdollisuuksia päästä käsiksi käyttäjien salasanoihin, mikäli hän pääsee tietokantaan käsiksi. Drupal 7:ssä salasanatietojen salaus on monivaiheista ja salasanat on käytännössä mahdotonta selvittää edes väsytyksen menetelmällä (engl. *bruteforce*). [50.]

Palvelimen muut tietoturvariskit on minimoitu jo palvelinkirjautumisen tasolla. Root-käyttäjänä ei pääse kirjautumaan, ja lisäksi SSH-yhteys on ohjattava DMZ-palvelimen tai VPN-yhteyden kautta.

Järjestelmän voi kuitenkin sotkea antamalla liikaa käyttöoikeuksia väärille käyttäjille.

Hätätilanteessa voidaan ottaa käyttöön tietokannan varmuuskopio, olettaen, ettei myös tiedostoja ole poistettu. Tietokannan varmuuskopio generoituu automaattisesti kerran päivässä.

6 Vertailu muihin saman alustan videojärjestelmiin

6.1 Omat projektit

Tapaus: raijaoranen.fi

Kirjailija Raija Oraselle vuonna 2007 luomani Drupal-sivusto (www.raijaoranen.fi) ei sisällä yhteisöllisiä ominaisuuksia eikä ole multimediatoiminnoiltaan kovin merkittävä. Sivustolla on kuitenkin muutamia videoita ja äänitteitä, jotka hyödyntävät pitkälti samoja tekniikoita ja moduuleja kuin StadiTV – vain paljon rajoittuneemmassa muodossa.

Kuvassa 9 näkyy sivustolla käytetty videosoitin, joka on varhaisempi versio StadiTV-projektin käyttämästä soittimesta.

Raija kertoo Leijonan osasta

► Leijonan osa



Videolla Raija kertoo uusimmasta romaanistaan: *Leijonan osa*.



"Ruusun aika" tv-sarjan tunnusvideo.

Kuva 9. Laajakuva- ja PAL-muotoisia progressiivisesti ladattavia videoita raijaoranen.fi-sivustolla.

Koska kyseessä oli edullinen jaettu palvelin (engl. *webhotel, shared hosting*) hyvin rajoitetuin resurssein, ei palvelimelle päässyt asentamaan mitään eikä sen ilmoitettu edes tukevan Drupal-järjestelmää. Drupal 5 kuitenkin saatiin lopulta toimimaan *php.ini*-muutoksilla. Muutettuja asetuksia olivat muun muassa *safe_mode*, *register_globals*, *memory_limit*, *max_execution_time* ja *AllowOverride*. Myöhemmin palveluntarjoaja muutti palvelinasetuksiaan niin, etteivät *php.ini*-muutokset enää toimineet, ja Drupal-sivusto muuttui käyttökelvottomaksi. Tämän sai kuitenkin korjattua suorittamalla muutokset *.htaccess*-tasolla *php.ini*:n sijaan. Samantapainen säätö on ollut tarpeen muissakin käyttämissäni jaetuissa palvelinympäristöissä.

Sivustolle asennettiin videoiden toistoon *flashvideo*-moduuli ja *JW FLV Player* – Flashsoitin, aivan kuten StadiTV-projektissakin. Palvelimella ei kuitenkaan ollut mahdollisuutta videoiden konversioon FFMPEG-ohjelmalla. Tämä ei ollut ongelma, koska videoita oli vain muutama ja päivittäjiä yksi. Flashvideo-asetuksista määritettiin, ettei moduuli yritä kääntää videoita, vaan videot muunnettiin flv-muotoon kääntämällä ne manuaalisesti ennen sivustolle siirtämistä. Tämä onnistuu Windowsissa esimerkiksi ilmaisella *WinFFMPEG*-ohjelmalla. Tähän projektiin nämä rajallisetkin toiminnot riittivät hyvin. Itse tutustuin samalla monimutkaisiin FFMPEG-asetuksiin, joita olen tarvinnut StadiTV:n lisäksi myös yksittäisissä videotuotannoissa.

Tapaus: Jublos

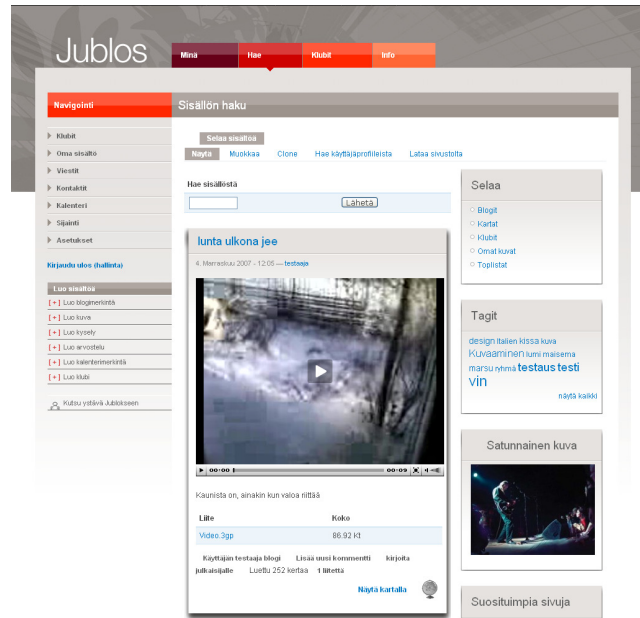
Jublos oli vuonna 2007 kehitysmielessä Drupal 5 -alustalle rakennettu yhteisöpalvelupilotti, joka sisälsi hyvin pitkälti samoja ominaisuuksia kuin StadiTV. Ominaisuudet keskittyivät jälleen yhteisöllisyyteen ja mobiilipuoleen, mutta myös automaattinen videojulkaisu oli tärkeässä roolissa.

Videoiden käsittelyn ja toiston hoitivat tässäkin projektissa *flashvideo*-moduuli, FFMPEG ja JW FLV Player. Käyttäjät pystyivät luomaan julkisia tai suljettuja ryhmiä ja lähettämään niihin videoita ja kuvia www-selaimella ja mobiililaitteilla. Mobiililaitella pystyi lähettämään videon useilla eri menetelmillä: puhelimen selaimelle optimoidulla www-sivulla, MMS-viestillä tai sähköpostitse.

Lisäksi tuotettiin tähän tarkoitukseen Symbian-pohjainen sovellus, jonka kehittäjänä toimi *Leonid Guzman*. Sovelluksen teko oli hyvin työlästä, erityisesti mobiililaitteen päässä, sillä tiedoston lähettämistä varten oli luotava oma implementaatio HTTP-protokollasta. Sovellus toimi valitettavasti vain vanhoilla *Symbian 2nd Edition* -puhelimilla, ja pienen muistin takia tiedostorajoitukset olivat varsin alhaisia.

Palvelulle toteutettiin useita testijaksoja, joihin kuuluivat muun muassa *Kehitysyhteistyö*-ryhmän virkistyspäivä, erään työntekijän polttarit ja *AavaCamp* eli kansainvälinen nuorten leiri Lapissa. Jälkimmäisessä testijaksossa nuorisoryhmälle annettiin viisi puhelinta rajoittamattomilla liittymillä ja he tuottivat parissa viikossa palveluun yli 500 multimediaviestiä (MMS). Viesteistä generoitiin sisältöä sivustolle kunkin käyttäjän ryhmään. Nuorten vanhemmille annettiin tunnukset, joilla he pääsivät seuraamaan nuorten tuottamaa yksityistä mediasisältöä. MMS-viestien käyttöasteesta voi päätellä, että se oli käyttökelpoisin käyttöliittymä. Valitettavasti MMS-viestien tekniset rajoitukset, kuten 300 kt:n maksimilähetys, pakottivat tuottamaan erittäin huonolaatuisia, lyhyitä videoita ja kuvia.

Kuvassa 10 näkyy esimerkki palvelun käyttöliittymästä ja siihen lähetetystä videosta.



Kuva 10. Jublos-palvelun videohakunäkymä.

Myöhemmin sain lisättyä palveluun paikkatietoisuuden, ja GoogleMaps-integraation, joiden avulla mediasisältöihin voitiin lisätä myös paikkatieto ja näyttää se kartalla. Elisalla oli mahdollisuus tuottaa myös jotain varsin ainutlaatuista: paikkatietojen tallennus SMS-viestillä suoraan palvelun kartalle. Tämä paikannus ei käyttänyt GPS-tietoa, vaan puhelimen tukimastojen sijaintien avulla laskettua summittaista triangulaatio-paikannusta.

Tekstiviestipaikannukseen olivat tekniikat jo olemassa, mutta niitä ei tietääkseni ollut käytetty Suomessa kaupallisessa mielessä – osittain varmasti lakirajoitusten takia. Paikannus yhdistettynä MMS-viestiin olisi voinut olla jopa merkittävä kaupallinen sovellus, mutta valitettavasti käyttämämme yhdyskäytävä (engl. *Gateway*) tuki paikannusta ainoastaan SMS-viesteille. En tiedä, onko järjestelmä myöhemmin kehittynyt tukemaan myös MMS-viestejä.

Jublos-palvelua ei jatkokehitetty eikä ylläpidetty pilottijakson jälkeen, joskin palvelu oli teknisesti käytettävissä vuoteen 2009 asti. Palveluun kehitettämäni mobiilitoiminnot olisi voitu monistaa myös StadiTV-projektin tarpeisiin, mutta StadiTV:ssä oli jo ennestään liikaa ominaisuuksia. Saavuttamani kokemus videoiden, yksityisten ja

julkisten ryhmien sekä karttaominaisuuksien osalta olivat kuitenkin varsin lupaavia, ja niitä hyödynnettiin myöhemmin StadiTV-projektissa.

Zircle

Zircle ei ole oma projektini, mutta Elisa julkaisi palvelun vuonna 2008. Palvelun ydin-ideana oli sama triangulaatiopohjainen paikannus kuin Jubloksessa, mutta se toimi nyt jatkuvana seurantana ilman erillisiä SMS-viestejä. En ollut itse rakentamassa palvelua, eikä se edes perustu Drupaliin, mutta liityin myöhemmin sen beta-käyttäjäksi. Palveluun kuului myös yhteisötoimintoja, joiden avulla saattoi seurata esimerkiksi ystäviensä tai kollegoidensa liikkeitä.

Zircle ei tietääkseni ole vielä saavuttanut kaupallista menestystä. Mahdollisesti syynä on vastaavantyyppinen GPS-pohjainen, yhteisöllinen, globaali paikannuspalvelu nimeltä *Google Latitude*. Näin siitäkin huolimatta, että *Zircle* ei vaadi käyttäjältä GPS-tietoja, ei datayhteyttä, eikä matkapuhelimeen asennettavaa sovellusta. Omien kokemuksieni perusteella *Google Latitude* tuhlaa akkua ja datakaistaa, mutta toisaalta se toimii kaikissa yleisissä dataliikennettä tukevissa liittymissä ja osaa hyödyntää GPS-tiedon lisäksi muita paikkatietoja, kuten WLAN-verkkoja. Lisäksi GPS-paikannus on noin metrin tarkkuudella paljon tarkempi kuin triangulaatiomenetelmällä, jonka tarkkuus mitataan sadoissa metreissä.

Espanja.org

Onnistuneen Jublos-pilotin jälkeen siinä tuotettuja mobiilitoimintoja päätettiin testata vuonna 2008 myös *www.espanja.org*- sivustolla. *Espanja.org* ei ole Elisan palvelu, vaan sitä ylläpitää epäkaupallinen yhdistys. Sain tehtäväkseni siirtää *Drupal 4* -pohjaisen sivuston *Drupal 5* -alustalle ja asentaa siihen kehittämäni SMS/MMS-viestien lähetyksen ja vastaanottojärjestelmän.

Tavoitteena oli testata mobiilitoimintoja mediasisältöjen luomiseen sivuston hyvin aktiivisella käyttäjäkunnalla. Päivityksen yhteydessä jouduin tekemään hyvin paljon työtä moduulien ja ulkoasun ajantasalle saattamisessa, erityisesti sen takia, että kyseessä oli jo vakiintunut tuhansien käyttäjien yhteisö ja sivuston piti pysyä käyttökunnossa.

Projektissa otettiin myös käyttöön modifioitu versio Jubloksen puhelinsovelluksesta, mutta se toimi vain harvoissa vanhoissa Symbian-puhelimissa, joita ei testaajilla ollut. Mobiililaitteille optimoitu www-sivu toimi kyllä yleisissä matkapuhelimissa, mutta sitä taas ei pidetty tarpeeksi helppokäyttöisenä tekniikasta tietämättömille.

Käytännön testitulokset jäivät hyvin laihoiksi: selvisi, etteivät juuri mitkään Espanjan operaattorit tukeneet MMS-viestien lähetystä. Ilmeisesti multimediatoinninnoille ei myöskään koettu olevan merkittävää tarvetta. Sen sijaan sivustolle asentamani Ajax-pohjainen chat-toiminto (*phpfreechat*) saavutti huomattavan suosion. Samanlaista toimintoa olisi voitu käyttää myös StadiTV:ssä tai muilla yhteisöllisillä sivuilla.

Case: TibeTube

TibeTube on Elisalle *Mika Peuralahden* kanssa kesällä 2009 toteuttamani videosivusto, joka liittyy varsin läheisesti StadiTV-projektiin. Kyseessä on *Lineage 2* -pelin laajennukselle omistetun sivuston video-osio. Projekti on muutaman työpäivän panoksella nopeasti toteutettu yksinkertainen Drupal 6 -asennus, joka hyödyntää sekä StadiTv:n videoarkkitehtuuria että samaa www-palvelinta. Käytännössä sivustolle on kopioitu flashvideo-moduuli ja Flash-soitin samoilla asetuksilla. Lisäksi aktivoitiin arvostelu-toiminto ja videoiden toisto ulkopuolisilla sivuilla (engl. *embedding*), mikä mahdollistaa videoiden jakamisen YouTuben tapaan.

Kuvassa 11 näkyy nykyaikaisella JW FLV Playerillä esitettävä video ja siihen liittyvät arvostelutoiminnot.

L2 Tiberius in the beggining-Joyful clan-



Embed: `<object class="display-flashvideo-mini" id="flashvid`

★★★★★★★★★★

Created by: popiella

Tags: funny

Kuva 11. TibeTube-palveluun liitetty peliaiheinen video.

Suurin osa projektiin käytetystä ajasta kului OpenID-integraation selvittämisessä. OpenID:lle on monenlaisia valmiita Drupal-ratkaisuja, mutta mikään niistä ei tuntunut tarpeeksi käyttäjäystävälliseltä. Ulkoasun osalta käytettiin valmista teemaa, jota ei jouduttu juurikaan jatkokäsittämään.

Projektin helppo ja pikainen toteutus kertovat StadiTV-alustan monistettavuudesta, mutta se, että sivusto oli asennettava samalla palvelimelle, osoittaa, ettei videokonversioiden keskittäminen yhdelle ja www-palveluiden erottaminen toiselle palvelimelle ole arkkitehtuurin kannalta onnistunut. Toisaalta siihen ei myöskään pyritty, koska tehokkaita palvelimia on hyvin rajallinen määrä käytettävissä ja toistaiseksi kumpikaan sivusto ei vaadi itsenäistä palvelinta.

6.2 Suomalaisia videopalveluita

Drupalilla on toteutettu lukuisia videopalveluita sekä Suomessa että ulkomailla. Kaikkia niistä on turha mainita, mutta Suomessa Drupal-alustaa hyödyntävät tällä hetkellä ainakin seuraavat Tv-kanavat:

- Yle
- Nelonen
- SuomiTV

Yle käyttää Drupalin multisite-ominaisuutta, jonka avulla toteutetaan toisistaan erillisiä alasivuja. Huomasin *YleAreenan* siirtyneen jossain vaiheessa Flash-videoiden käyttöön aiemman WMV-muodon sijaan. YleAreena vaikutti myös siirtyneen Drupalista hiljattain *Zend Frameworkin* päälle. Vieraillessani *DrupalCamp Helsinki* -tapahtumassa marraskuussa 2009 sain tilaisuuden kysyä Ylen edustajalta syytä tähän. Vastaus oli hieman epämääräinen ilmoitus, ettei kyse ollut Drupalin puutteista, mutta palvelu haluttiin nyt rakentaa toisella tavalla – vaikka muu sivusto pohjautuikin Drupaliin. [38.]

Samassa DrupalCamp-tapahtumassa oli myös mielenkiintoinen luento Nelonen-kanavan Drupal-toteutuksesta. Huomiota herätti erityisesti kuvaus järjestelmän vahvasta muokkauksesta skaalautuvuuden varmistamiseksi. Esimerkiksi olemassa olevia Drupalin äänestysmoduuleja ei hyväksytty sellaisenaan käyttöön, vaan ne kirjoitettiin osittain uusiksi tehokkuuden parantamiseksi.

Lisäksi sivusto käyttää käänteistä välityspalvelinta nimeltä *Varnish* lähes kaiken sisällön näyttämiseen, mikä toimii hyvin, koska kävijät ovat anonyymejä eikä heille tarvitse tuottaa kovin dynaamista sisältöä. Muutamat toiminnot, jotka saavuttavat hetkellisesti valtavia käyttäjämääriä, on toteutettu pääsivustosta täysin erillisinä. Näin varmistetaan,

ettei yleisöryntäys sekoita pääpalvelua. Nelonen ei käytä videotiistossa Brightcove-järjestelmää, vaan ilmeisesti jotakin sisäisesti tuotettua Flash-soitinta. [38; 40.]

Uusimman suomalaisen tv-kanavan, *SuomiTV:n*, www-sivusto rakennettiin alusta asti Drupalille. Osallistuin itse palvelun toteuttamiseen uudessa työpaikassani.

Osallistumispanokseni oli kuitenkin verrattain pieni muiden ensisijaisten projektieni takia, enkä ollut aktiivisesti tekemässä sivuston myöhemmin toteutettua NettiTV-osuutta.

Videojärjestelmän raskaimmat osat on osittain ulkoistettu *Brightcove*-palvelulla ja integroitu Drupaliin muun muassa *media*- ja *feeds*-moduulien sekä useiden kustomoitujen moduulien avulla. Tämä ratkaisu mahdollisti NettiTV-osuuden ennätysnopean pystyttämisen. Itse pidän videototeutusta muuten hyvin onnistuneena, mutta koko ruudun videotilaan (engl. *fullscreen*) siirtyminen aiheuttaa harmittavan pysähdyksen videotiistossa, ilmeisesti kaistanleveystemuutoksen takia. Useimmissa muissa järjestelmissä käytetään nähtävästi samaa videoversiota myös koko ruudulla näytettävälle videolle, jolloin videotiisto ei häiriinny. [51.]

7 Yhteenveto

Insinööriyöprojektini ei edennyt aivan suunnitellulla tavalla: moni StadiTV:n suunnitelluista ominaisuuksista toteutui vain osittain. Syyksi voitaisiin esittää projektin laman minimoimaa budjettia, liian kunnianhimoisia visioita, työnantajani vaihtumista tai sitten vain omia puutteitani. StadiTV-projekti on kuitenkin pitkän tähtäimen projekti, joten on mahdollista, että projektia vielä jatketaan muiden tahojen toimesta.

Omana tavoitteenani oli osoittaa insinööriyölläni myös Drupalin mahdollisuudet teknisesti haastavien www-projektien sovellusallustana. Nähdäkseni se on maailmanlaajuisesta suosiostaan vielä lähes tuntematon mediatekniikan koulutusohjelmassa, siitäkin huolimatta että Drupal-kehittäjille on tällä hetkellä erittäin kova kysyntä. Uskon, että koko projektia ei olisi voitu edes lähteä kehittämään ilman Drupal-alustaa tai muita avoimia työkaluja, kuten FFmpeg:ää, x264:ää ja muita LAMPP-arkkitehtuurin sovelluksia.

Monipuolisen, yhteisöllisen videojulkaisujärjestelmän voi toteuttaa monin tavoin, ja hyvin rajoitettu budjettikaan ei ole välttämättä este. Monikanavainen palvelu, joka perustuu yhteiseen sisältölähteeseen, on myös toteutettavissa, mutta resurssien tarve kasvaa nopeasti, koska kunkin alustan rajoitukset ja mahdollisuudet on ymmärrettävä ja niihin on sopeuduttava. On varsin vaikea kehittää palvelua tietyllä päätelaitteelle, kun päätelaitteet ovat vielä hyvin harvinaisia ja huonosti tuettuja. Toistaiseksi ainakin DVB-H-signaalia toistavat mobiili-tv-puhelimet kuuluvat tähän luokitteluun.

Insinööriyöraporttia tehtäessä todettiin, että tekniikoiden kehittyessä joutuvat myös nykyisin johtavat teknologiat todennäköisesti väistymään uusien tieltä. Suljettujen Flash- ja H.264-standardien rinnalle ovat nousemassa avoimet ja maksuttomat standardit, kuten HTML5 ja Googlen VP8-videokoodekki. Google toimii todennäköisesti suunnannäyttäjänä suosituimpien tekniikoiden syrjäytysprosessissa. StadiTv-projektin toteutuksessa oli kuitenkin käytettävä nykyisiä standardeja.

Vastaavien projektien suunnitteluissa pitäisi kuitenkin löytää projektista oleelliset tavoitteet ja koettaa pysyä niissä, erityisesti mikäli resursseja on hyvin rajallisesti. En voi väittää osaavani toimintojen keskittämistä erityisen hyvin, vaan olen tottunut vastaamaan kaikkiin ominaisuuspyyntöihin ”voidaan toteuttaa”. Vaikka teknisesti melkein kaikki on mahdollista, lopputulos riippuu aina ajasta ja resursseista. Työelämässä on osattava arvioida realistisesti, milloin tehdä kompromisseja ja jättää ominaisuuksia pois. Tämän arviointikyvyn saavuttaminen vaatii paljon kokemusta.

StadiTV:n teknisen toteutuksen keskipisteessä olevan Drupal-järjestelmän todettiin soveltuvan hyvin videopalvelun integraatioalustaksi. Todellista testiä, eli skaalautuvuutta suurten käyttäjämäärien rasitteen alla, ei työn aikana päästy kokemaan. Vertailukohteena olleet esimerkit todistavat alustan kuitenkin tarvittaessa myös skaalautuvan tehokkaasti.

StadiTV-palvelun ydinteemat eli monikanavaisuus, paikallisuus, multimedia ja yhteisöllisyys, eivät ole vain hetkellisiä trendejä. Vaikka näille kaikille löytyy ennestään palvelutarjontaa, ei kaikkia nähdäkseni ole vielä yhdistetty tehokkaasti. Tosin aika lähellä ollaan jo: esimerkiksi YouTube ja erityisesti Facebook ovat integroitavissa lähes kaikkeen. Näiden erittäin suosittujen palveluiden ja lukuisten muiden kilpailijoiden täyttämällä videojulkaisun ja sosiaalisen median kentällä on todennäköisesti olemassa vielä jotain markkinarakoja.

Lähteet

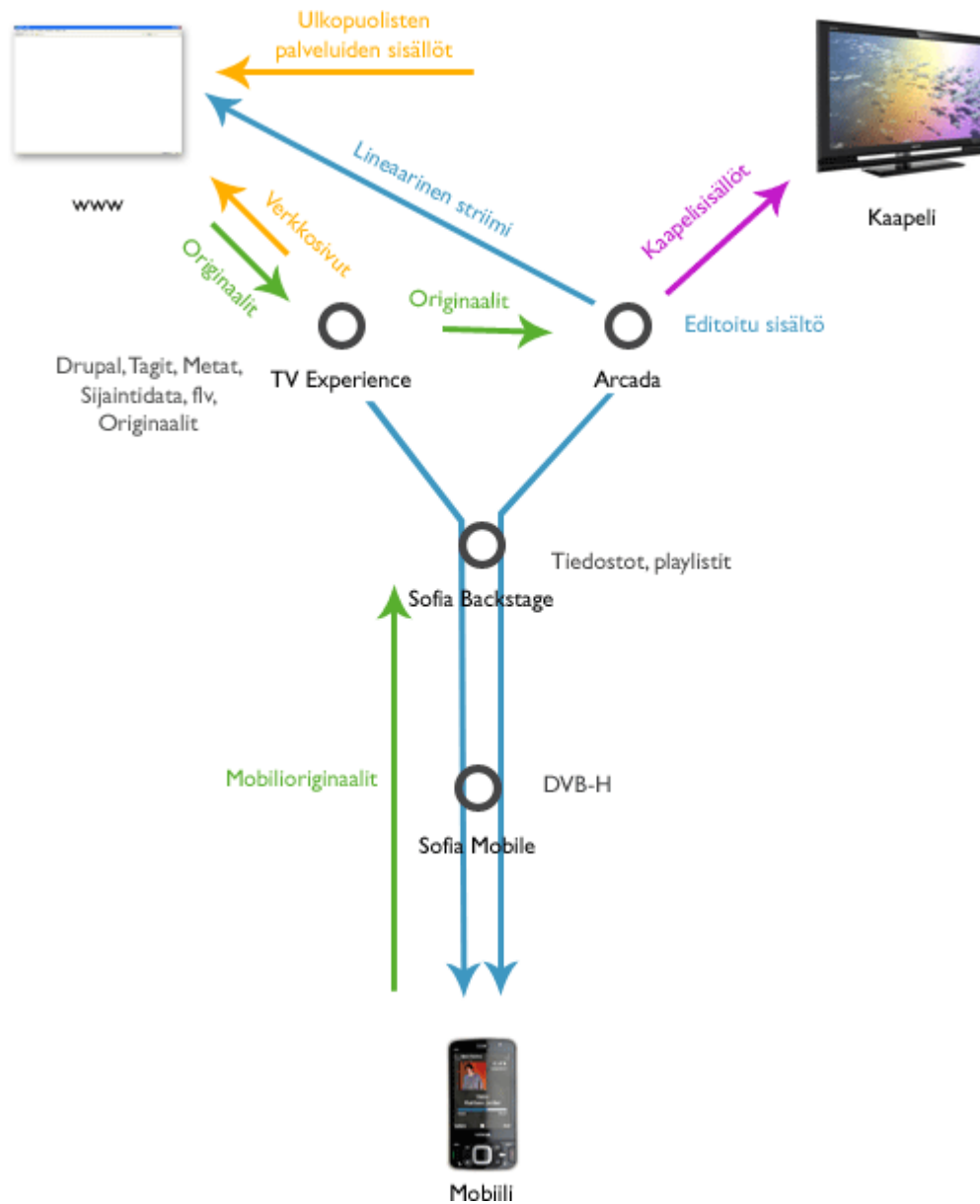
1. The Rise of Flash Video, Part 1. (WWW-dokumentti.) Digital Web Magazine.
<http://www.digital-web.com/articles/the_rise_of_flash_video_part_1>. Päivitetty 9.9.2006.
Luettu 25.4.2010.
2. Windows Media Video. (WWW-dokumentti.) Wikipedia.
<http://en.wikipedia.org/wiki/Windows_Media_Video>. 13.4.2010. Luettu 28.4.2010.
3. Video codec. (WWW-dokumentti.) Wikipedia. <http://en.wikipedia.org/wiki/Video_codec>.
29.4.2010. Luettu 25.4.2010.
4. Audio-video patents. (WWW-dokumentti.) Software patents Wiki.
<http://en.swpat.org/wiki/Audio-video_patents>. 31.3.2010. Luettu 20.4.2010.
5. H.264/MPEG-4 AVC. (WWW-dokumentti.) Wikipedia.
<http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC>. 28.4.2010. Luettu 29.4.2010.
6. x264 - a free h264/avc encoder (WWW-dokumentti.) VideoLAN-project.
<<http://www.videolan.org/developers/x264.html>>. Luettu 12.5.2010.
7. Comparison of video services. (WWW-dokumentti.) Wikipedia.
<http://en.wikipedia.org/wiki/Comparison_of_video_services>. 29.4.2010. Luettu 30.4.2010.
8. Blodget, Henry. YouTube Can Be Very Profitable In The Near Future. (WWW-dokumentti.)
Business Insider. <<http://www.businessinsider.com/henry-blodget-youtube-can-be-very-profitable-in-the-near-future-2009-7>>. 27.7.2009. Luettu 26.4.2010.
9. Google Trends: youtube, "google video", metaface, dailymotion, youku. (WWW-dokumentti.)
Google Trends. <<http://www.google.com/trends?q=youtube,+%22google+video%22,+metaface,+dailymotion,+youku&ctab=0&geo=all&date=all&sort=0>>. 28.4.2010.
10. YouTube. (WWW-dokumentti.) Wikipedia. <<http://en.wikipedia.org/wiki/YouTube>>.
28.4.2010. Luettu 29.4.2010.
11. US Broadband Penetration Grows to 95.3% Among Active Internet Users - March 2010
Bandwidth Report. (WWW-dokumentti.) <<http://www.websiteoptimization.com/bw/1003/>>.
31.1.2010. Luettu 27.4.2010.
12. Flash Video. (WWW-dokumentti.) Wikipedia. <http://en.wikipedia.org/wiki/Flash_Video>.
28.4.2010. Luettu 30.4.2010.
13. Summary of AVC/H.264 license terms. (WWW-dokumentti.) MPEGLA.
<http://www.mpegla.com/main/programs/AVC/Documents/AVC_TermsSummary.pdf>.
25.3.2010 (PDF-metadatan mukaan).
14. Lawler, Ryan. Google to Open-source VP8 for HTML5 Video. (WWW-dokumentti.)
NewTeeVee. <<http://newteevee.com/2010/04/12/google-to-open-source-vp8-for-html5-video/>>.
12.4.2010. Luettu 28.4.2010.

15. Dignan, Larry. A look at On2 Technologies and why Google wants it. (WWW-dokumentti). ZDNet. <<http://www.zdnet.com/blog/btl/a-look-at-on2-technologies-and-why-google-wants-it/22242>>. 5.8.2009. Luettu 12.5.2010.
16. Blizzard, Christopher. HTML5 video and H.264 – what history tells us and why we’re standing with the web. (WWW-dokumentti). Blogi. <<http://www.0xdeadbeef.com/weblog/2010/01/html5-video-and-h-264-what-history-tells-us-and-why-were-standing-with-the-web/>>. 24.1.2010. Luettu 24.4.2010.
17. Wauters, Robin. Google Acquires Video Compression Technology Company On2 For \$106 Million. (WWW-dokumentti). TechCrunch. <<http://techcrunch.com/2009/08/05/google-acquires-video-compression-technology-company-on2-for-106-million/>>. 5.8.2009. Luettu 30.4.2010.
18. YouTube HTML5 Video Player. (WWW-dokumentti). YouTube. <<http://www.youtube.com/html5>>. 27.1.2010. Luettu 25.4.2010.
19. HTML5. (WWW-dokumentti.) Wikipedia. <<http://en.wikipedia.org/wiki/HTML5>>. 28.4.2010. Luettu 29.4.2010.
20. Real Time Messaging Protocol. (WWW-dokumentti.) Wikipedia. <http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol>. 28.4.2010. Luettu 29.4.2010.
21. Manual Wowza Flash Streaming Server. (WWW-dokumentti.) Rambla Wiki. <http://rampubwiki.wiki.rambla.be/Manual_Wowza_Flash_Streaming_Server>. 17.7.2009. Luettu 27.4.2010.
22. Buytaert, Dries. Why PHP (and not Java). (WWW-dokumentti). Blogi. <<http://buytaert.net/why-php-and-not-java>>. 27.4.2006. Luettu 20.4.2010.
23. Buytaert, Dries. Java.net using Drupal. (WWW-dokumentti). Blogi. <<http://buytaert.net/java-net-using-drupal>>. 8.9.2006. Luettu 20.4.2010.
24. Programming Language Popularity. (WWW-dokumentti). DedaSys LLC. <<http://www.langpop.com>>. 22.4.2010. Luettu 28.4.2010.
25. Red Hat Enterprise Linux. (WWW-dokumentti.) Wikipedia. <http://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux>. 30.4.2010. Luettu 30.4.2010.
26. System requirements. (WWW-dokumentti.) Drupal.org. <<http://drupal.org/requirements>>. 8.4.2010. Luettu 10.11.2009.
27. nGinx, kevyt webpalvelin. (WWW-dokumentti.) Wikipedia. <<http://en.wikipedia.org/wiki/Nginx>>. 30.4.2010. Luettu 30.4.2010.
28. April 2010 Web Server Survey. (WWW-dokumentti.) Netcraft. <http://news.netcraft.com/archives/web_server_survey.html>. 15.4.2010. Luettu 20.4.2010.
29. Tuntematon tekijä. Neo Arena TVExperience System Documentation. Luottamuksellinen asennusohje. Oulun seudun ammattikorkeakoulu/NeoArena. 2009.
30. Darwin Streaming Server. (WWW-dokumentti.) Wikipedia. <http://en.wikipedia.org/wiki/Darwin_Streaming_Server>. 22.9.2009. Luettu 10.4.2010.

31. PHP Performance tip: require versus require_once. (WWW-dokumentti.) Tech Your Universe. <http://www.techyouruniverse.com/software/php-performance-tip-require-versus-require_once>. 1.8.2008. Luettu 10.4.2010.
32. PHP: Runtime Configuration. (WWW-dokumentti.) The PHP Group. <<http://php.net/manual/en/apc.configuration.php>>. 30.4.2010. Luettu 5.5.2010.
33. Can a Drupal web site handle a million page views a day? (WWW-dokumentti.) 2bits. <<http://2bits.com/articles/can-a-drupal-web-site-handle-a-million-page-views-a-day.html>>. 13.3.2008. Luettu 10.4.2010.
34. Benchmarking Drupal with PHP op-code caches: APC, eAccelerator and XCache? (WWW-dokumentti.) 2bits. <<http://2bits.com/articles/benchmarking-drupal-with-php-op-code-caches-apc-eaccelerator-and-xcache-compared.html>>. 20.4.2008. Luettu 10.4.2010.
35. Memcached. (WWW-dokumentti.) Google Project Hosting. <<http://code.google.com/p/memcached/wiki/NewOverview>>. 10.4.2010. Luettu 12.5.2010.
36. Network File System (protocol). (WWW-dokumentti.) Wikipedia. <[http://en.wikipedia.org/wiki/Network_File_System_\(protocol\)](http://en.wikipedia.org/wiki/Network_File_System_(protocol))>. 30.4.2010. Luettu 30.4.2010.
37. Symbolinen linkki. (WWW-dokumentti.) Linux.fi / Wiki. <http://linux.fi/wiki/Symbolinen_linkki>. 1.6.2009. Luettu 20.10.2009.
38. Useita luennoitsijoita. 2009. DrupalCamp Helsinki. Drupal-aiheinen seminaari, 27.11.2009.
39. Kieran, Lal. Whitehouse.gov re-launches on Drupal and engages the Drupal community at DC users meeting. (WWW-dokumentti.) Drupal.org. <<http://drupal.org/whitehouse-gov-launches-on-drupal-engages-community>>. 18.11.2009. Luettu 20.11.2009.
40. Drupal Camp 29.11.2009. (WWW-dokumentti.) Blogi. <<http://blogit.otavanopisto.fi/tekno/2009/11/30/drupal-camp-29-11-2009/>>. 30.11.2009. Luettu 10.4.2010.
41. Kieran, Lal. Drupal Wins Best Overall 2008 Open Source CMS Award for Second Year in a Row. (WWW-dokumentti.) Drupal.org. <<http://drupal.org/Drupal-Wins-Best-Overall-2008-Open-Source-CMS-Award-Packt>>. 31.10.2008 päivitetty. Luettu 1.11.2008.
42. Buytaert, Dries. Next steps for Drupal 7 usability. (WWW-dokumentti.) Drupal.org. <<http://drupal.org/node/474338>>. 27.5.2009. Luettu 5.4.2010.
43. MySQL. (WWW-dokumentti.) Wikipedia. <<http://en.wikipedia.org/wiki/MySQL>. 30.4.2010>. Luettu 30.4.2010.
44. Oracle's Takeover of Sun Microsystems May Impact Java, MySQL and other Open Source Apps. (WWW-dokumentti.) Technology Talks. <<http://www.kokeytechnology.com/internet-technology/oracle%E2%80%99s-takeover-of-sun-microsystems-may-impact-java-mysql-and-other-open-source-apps/>>. 19.3.2010. Luettu 10.4.2010.
45. YouTube Architecture. (WWW-dokumentti.) Possibility Outpost. <<http://highscalability.com/youtube-architecture>>. 12.3.2008. Luettu 15.4.2010.

46. Zhao, Haiping. HipHop for PHP: Move Fast. (WWW-dokumentti.) Facebook. <<http://developers.facebook.com/blog/post/358>>. 2.2.2010. Luettu 10.3.2010.
47. FFmpeg Documentation. (WWW-dokumentti.) FFmpeg.org. <<http://ffmpeg.org/ffmpeg-doc.html>>. 29.4.2010. Luettu 30.4.2010.
48. Moisio, Aleksi. Floobs hakeutuu konkurssiin. (WWW-dokumentti, artikkeli). Digitoday. <<http://www.digitoday.fi/bisnes/2010/01/21/floobs-hakeutuu-konkurssiin/20101000/66>>. 21.1.2010. Luettu 10.4.2010.
49. Krug, Steve. Älä pakota minua ajattelemaan - Tervettä järkeä verkkosuunnitteluun. Jyväskylä: Readme.fi, 2006.
50. Demaria, Tiziano. Password hashing question. (WWW-dokumentti.) Drupal.org. <<http://drupal.org/node/29405>>. 24.1.2009>. Luettu 15.4.2010.
51. SuomiTV:n NettiTV ennätysajassa. (WWW-dokumentti, artikkeli.) Kauppalehti. <<http://www.kauppalehti.fi/5/i/yritykset/lehdisto/hellink/tiedote.jsp?selected=kaikki&oid=20100401/12706241441790>>. 7.4.2010. Luettu 7.4.2010.

Liite 1: Multim mediasisältöjen käsittelyvaiheet



Käyttäjän tuottaman sisällön käsittelyvaiheet. Toteutus Kimmo Tukiainen, Forum Virium.

Liite 2: Luettelo käytetyistä Drupal-moduuleista

Alla näkyvät StadiTV-projektin kaikki aktiiviset moduulit. Tärkeimpien moduulien nimet ovat paksunnettuna. Moduuleista saa lisätietoa osoitteessa [http://drupal.org/project/\[moduulin_nimi\]](http://drupal.org/project/[moduulin_nimi])

Listaus haettiin tietokannasta MySQL-käskyllä: `SELECT `name` FROM `system` WHERE STATUS = 1 AND TYPE = 'module' ORDER BY `system`.`filename` ASC`

Kustomointimoduulit (custom)

staditv

Ydinmoduulit (core)

comment

database_logging

help

locale

menu

path

php_filter

search

statistics

syslog

taxonomy

trigger

update_status

upload

(lisäksi pakolliset: block, filter, node, system, user)

Laajennusmoduulit (contributed)

admin_menu

advanced_help

backup_migrate

browscap

calendar

jcalendar

captcha

image_captcha

content

content_copy

content_permissions

fieldgroup

nodereference

number

optionwidgets

text

userreference

computed_field

date

date_api

date_timezone

deadwood

devel

emaudio

emimport

emthumb

emvideo

emfield

flashvideo

flashvideo_cck

flashvideo_streaming

flashvideo_views

geonames

geonames_tools

gmap

gmap_location

gmap_macro_builder

image_attach

image_im_advanced

image

link

location_cck

location_search

location

location_node

location_user

mimedetect

mobile_theme

poormanscron

prepopulate

xmlrpc_server

services

tac_lite

transliteration

module_weights

system_module

util

views

views_export

views_ui

vote_up_down

votingapi

Liite 3: Sivuston käyttöliittymän eri versiot

Mobiiliversio Ohje


Testi Käyttäjä Kirjaudu Ulos

Etusivu Ryhmät Omat videot Lisää video


Hae omia videoita Tarkennettu haku

Hae


Lorem ipsum dolor sit amet, consectetur adipiscing




Lorem ipsum dolor sit amet, consectetur adipiscing




Lorem ipsum dolor sit amet, consectetur adipiscing



Lorem ipsum dolor sit amet, consectetur adipiscing




Lorem ipsum dolor sit amet, consectetur adipiscing



Lorem ipsum dolor sit amet, consectetur adipiscing




Lorem ipsum dolor sit amet, consectetur adipiscing



Lorem ipsum dolor sit amet, consectetur adipiscing

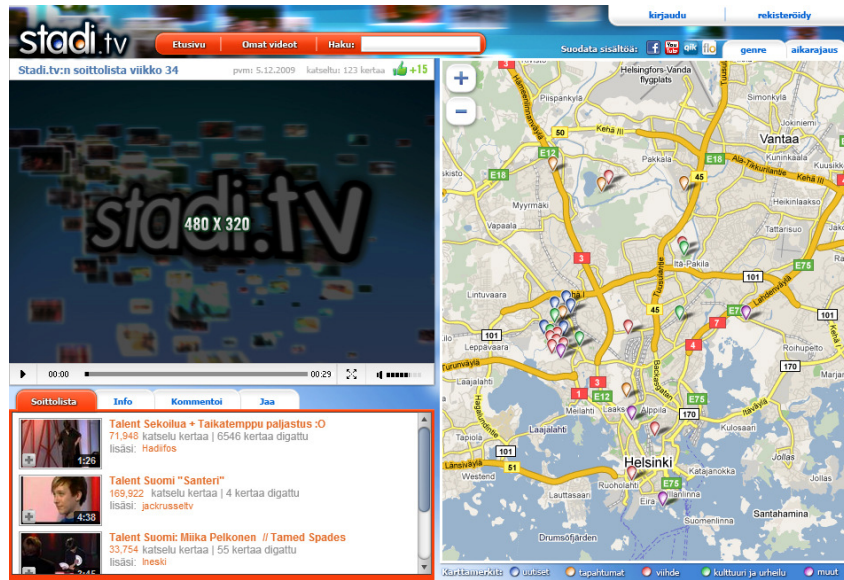


Lorem ipsum dolor sit amet, consectetur adipiscing

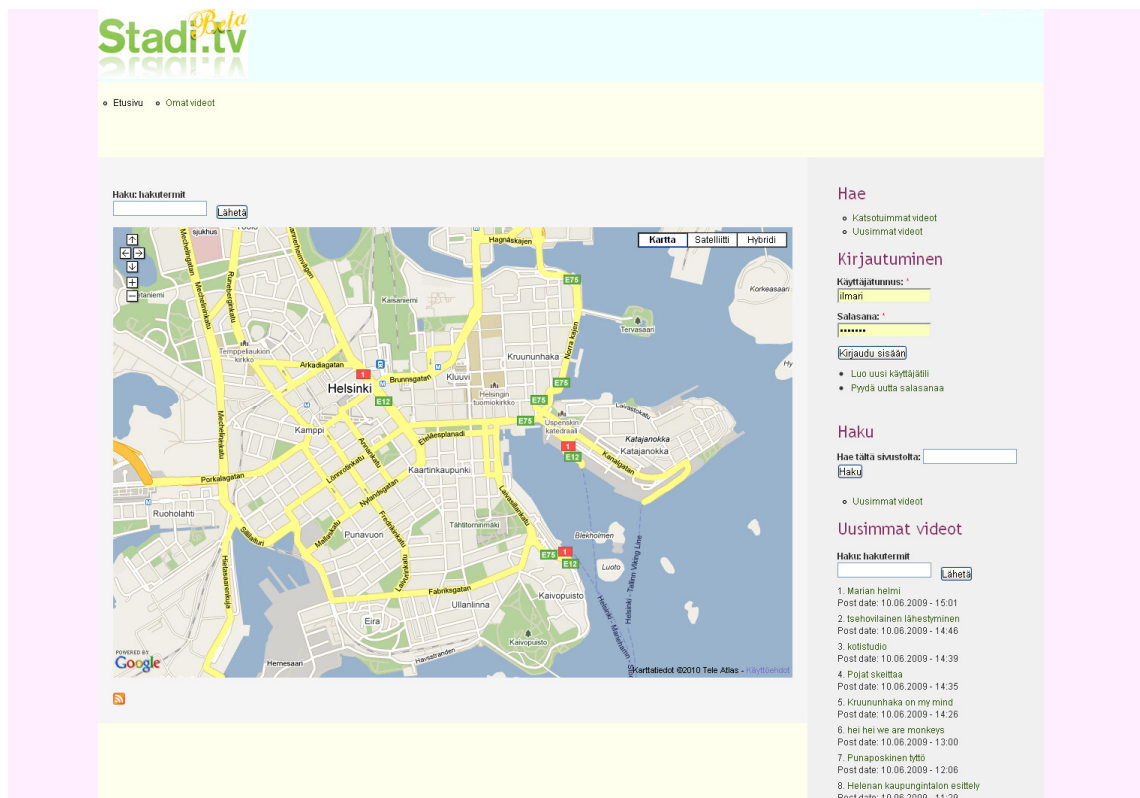


Alkuun | Edellinen | **[1]** 2 3 4 5 | Seuraava | Loppuun

Käyttöliittymä 1: tyylikäs ja vähäeleinen mutta toiminnoiltaan puutteellinen.




Käyttöliittymä 2a: Graafikon suunnittelema toimivan oloinen ulkoasu.



Käyttöliittymä 2b: Wireframeversio, jonka pohjalta pyritään lähestymään graafikon näkemystä.

Käyttäjien hallinta
Raportit
Advanced help
Ohje



Käyttäjät:
Kirjaudu ulos

Etusivu
Dokumentaatio
Omat videot
Hae tätä sisältöä:
Haku


Golffi keittää

Näytä
Äänestystiedot
Muokkaa
Seuranta
Dev load
Dev render

Kirjoittanut admin 10 Kesäkuu, 2009 - 05:22

tagit: golf keittää keittää

Auton lämmityslaitteita ruiskuttamaan vettä suoraan moottoriin vähän ennen kehä ill:n ja Turun moottorin risteystä. Auto oli juuri katsastettu edellisenä päivänä. Pääsin sentään kotiin asti ja sain vaihdettua letkun (melkein ihan itselti).



Lisää uusi kommentti: 1 piste

- Oma käyttäjälini
- Luo sisältöä
- Dokumentaatio
- Kanava
- Kuva
- Oma video
- Online-video
- Kirjaudu ulos

Käyttöliittymä 2c: keskeneräinen ulkoasun toteutus.